ESD-TR-65-276

AN APPROACH TO COMPUTER INSTALLATION PERFORMANCE EFFECTIVENESS EVALUATION

ESLB

N. Statland
R. Proctor
J. Zelnick
et al

JUNE 1965

(Directorate of Analysis, Deputy for Advanced Planning, Electronic Systems Division, Air Force Systems Command, L. G. Hanscom Field, Bedford, Massachusetts, 01731.)

ESD-TR-65-276

AN APPROACH TO COMPUTER INSTALLATION PERFORMANCE EFFECTIVENESS EVALUATION

N. Statland
R. Proctor
J. Zelnick
et al

JUNE 1965

(Directorate of Analysis, Deputy for Advanced Planning, Electronic Systems Division, Air Force Systems Command, L. G. Hanscom Field, Bedford, Massachusetts, 01731.)

FOREWORD

This report has been published for Development Planning Study 7990, Task 38, by the AUERBACH Corporation, 1634 Arch Street, Philadelphia, Pennsylvania 19103. It covers work performed under contract AF19(628)-5035 and the document bears AUERBACH report number 1243-TR-2. The Air Force Program Monitor was Mr. William J. Granacki, ESLB.

The contract was originally based upon the work performed by AUERBACH Corporation under contract AF19(628)-2838 which led to the development of the VECTOR process for estimating computer processing time. The research reported in this document was conducted from December 1964 through April 1965 under the direction of Mr. Norman Statland as Program Manager.

The complete list of authors is as follows:

N. Statland
R. Proctor
J. Zelnick
R. Getis
J. Anderson
A. Taylor

This technical report has been reviewed and is approved.

William J. Granacki

WILLIAM J. GRANACKI
Project Officer

Francis J. Dillon Jr

FRANCIS J. DILLON, JR.
Colonel, USAF
Director of Analysis
Deputy for Advanced Planning

ABSTRACT

Decisions regarding the implementation and evaluation of data processing systems have reflected costly compromises selected from many possible alternatives and there is a need for guidelines which can lead to better decisions regarding automation utilization and planning. The AUERBACH process for computer installation performance effectiveness evaluation operates on a set of specifications and characteristics regarding the principal problem tasks of an installation, its computer complex, and administrative and financial performance.

The process provides objective measures of performance efficiency based on both quantitative and qualitative data, and provides standards for measuring installation effectiveness. Specifications and characteristics are collected via questionnaires, once and only once, in four categories: computer hardware, extended machine (hardware/software interaction), software evaluation and problem specification.

Computer problems can be classified by the environments in which they function, the sources from which data is received and its implied sequence, and the response time within which the computer system is to react. Significant hardware and problem characteristics can be identified, as demonstrated in the AUERBACH VECTOR process (see ESD-TDR-64-194) and estimated running time computed. An extension of this measurement of computer system performance provides a rating for the performance of a given software package on a given piece of hardware by comparing the time derived from the hand-tailored coding to the timing resulting from the object program produced by the software. This ratio measures the efficiency of the software on the specific hardware configuration. The aggregate ratios for all the individual performance criteria are used to derive a performance standard for a software system.

Algorithms are used to summarize the raw data elements and a computer program will select data elements, make simple arithmetic combinations of these elements into composites, and prepare the data for entry into a statistical analysis. Stepwise multiple regression analysis is utilized to determine the relative significance of various data elements and to calculate their relative weights.

TABLE OF CONTENTS

TABLE OF CONTENTS (Contd.)

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

1.1 STATEMENT OF THE PROBLEM

The electronic computer has had a most significant effect upon the conduct of U.S. Air Force activities. Use of computer equipment has enabled the Air Force to carry out programs never before possible, and has facilitated the provision of more effective and economic services. The extremely rapid exploitation of the computer evidenced in the Air Force has not been without problems involving aspects of acquisition and utilization. Some examples of the problem areas are:

(1) The absence of precise, objective measurement criteria for the selection of new equipment and evaluation of existing installations.

(2) The absence of policies and operational guidelines that can be applied uniformly to all Air Force ADP activities.

(3) The absence of measurement criteria designed to account for the effect that overall systems design has on the efficiency and effectiveness of computer operations.

(4) The absence of measurement criteria to determine those applications which produce distinct advantages as opposed to others where advantages are marginal at best.

A procedure is needed that:

(1) Is not overly expensive.

(2) Will lead to the right choice of equipment.

(3) Will make available to policy management a set of evaluation criteria which can be applied to all Air Force installations or to homogeneous subsets of installation types in order to determine guidelines for effective spending of dollars.

This procedure must measure computer system performance in terms of rigid system requirements, software, effectiveness of the installation management in applying its resources to on-time delivery of required outputs, and getting the job done at minimum cost and with maximum programming efficiency.

1.2 BACKGROUND

In the past few years, some work has been done in the area of effectiveness evaluation; the Bibliography in this report is indicative of the nature of this work. Two efforts in this direction are described below.

1.2.1 AUERBACH Standard EDP Reports

As a consulting organization specializing in problems of information technology, AUERBACH Corporation has encountered many client problems associated with the evaluation of computer systems performance. The experience the corporation gained in performing evaluations for client companies and government agencies led to the first efforts to provide a comprehensive standard performance evaluation. These efforts resulted in the production of the AUERBACH Standard EDP Reports (ASEDPR) [1], an eight-volume encyclopedic set of computer specifications, software characteristics, prices, and performance evaluations on standard problems.

The ASEDPR approach to comparative equipment evaluation is to define equipment configuration parameter sets for a wide range of installation configurations and obtain timing data as a measure of their performance on standard benchmark problems. The performance estimates are obtained by actually coding critical portions (later referred to as macrofunctions) of the problems. The timing information and evaluation data on the computer system are sent to the manufacturer involved to check errors of fact before publication.

Standard EDP Reports represented a major step toward providing industry-wide useful information for comparing various equipments. They permit potential users who anticipate acquiring data processing equipment to compare, from among the various offerings of the manufacturers, the performance of similar configurations on standard problems. The use of the performance curves assumes that the potential user is able to find a configuration that is similar to the one he contemplates, and a benchmark problem that typifies his application(s).

1.2.2 VECTOR Process

Concern with how to sharpen the precision of equipment performance measurement in a particular application situation and in a particular configuration led to

---

(1) AUERBACH Standard EDP Reports, AUERBACH INFO, INC.

the concept of providing independent descriptions of hardware, (i.e., central processors, I/O channels, peripheral devices, etc.), independent descriptions of particular problems, and a means of combining these independently derived description-measures to produce a particular performance measure for a problem on a particular configuration of hardware. This concept was developed by AUERBACH under contract AF 19(628)-2838 for the U.S. Air Force, and led to the specification of the VECTOR process.[2]

The VECTOR process validated the conceptual approach, and demonstrated the feasibility of combining two independently derived detailed specifications, one hardware and one problem descriptive, to produce performance estimates of high quality. The technique was validated against the performance estimates arrived at by the ASEDPR for selected problems, and showed high correlation with these results.

In development of the VECTOR technique, the factors that most influence the performance of the equipment for a class of problems were identified. Specifications for both hardware characteristics and problem characteristics were developed, refined, combined, and refined again to arrive at the final sets of computer and problem characteristics of importance. Elements such as add time, cycle time, multiply and divide time, etc., that "apparently" had large influence were found to be of little importance, while seemingly insignificant factors such as code and radix conversion, and pack and unpack facilities, were found, upon further analysis, to play important roles in determining computer performance.

1.3 APPROACH

Performance effectiveness measures are made up of many elements, some of which are easily quantified and others which can only be specified in qualitative terms. Examples of quantified measures are seen in the time to perform $A + B \rightarrow C$, or the time to perform the same operation using a software system on the same hardware configuration (which interaction we have labelled the extended machine), the cost of the equipment, and the number of programmers working in the installation.

---

(2) Taylor, A., Hillegass, J.R., and Statland, N., Quantitative Methods for Information Processing Systems Evaluation, ESD-TDR-64-194, AUERBACH Corp., January 1964.

More difficult to specify as elements contributing to performance effectiveness are such items as the value of the software in speeding up the production of a task, the quality of the programming, the efficiency of the console and peripheral equipment operators plus the effectiveness of the managers in allocating such resources as programmers, operators, supplies, machine time, clerical support, and equipment capacity.

The previous work enabled the AUERBACH study team to rapidly attack the concept of the performance measurement procedure. As a point of departure for the study, it was decided to base part of our approach upon the work we had already done in the VECTOR technique as a generalized means of measuring performance, augmenting it with the factors necessary to measure and integrate software into the performance evaluation, and to apply a similar methodology to other areas of operational significance.

1.3.1 Methods of the Evaluation Procedure

The use of the term performance measurement was construed to take into account the following classifications of factors affecting the overall performance of computer installations:

(1) The equipment configuration, its hardware performance characteristics expressed in speed and storage capacities of central processor and equipment, plus performance times of the central processor on problem-oriented macrofunction loops.

(2) The interaction, expressed in modified central processor performance times, of a specific software system on the hardware in the form of a modified or extended machine.

(3) The definition of the system requirements in the form of specification of the definitive characteristics of sets of representative problems.

(4) The functional classification of installation environments to determine the proper set of representative problems to be used as the basis for measuring programming effectiveness as well as enabling management of resources to be measured by standards derived under varied environmental constraints (e.g., on-line, open-shop, integrated operations, central computing services or closed-shop, etc.).

(5) A set of software specifications designed to qualitatively evaluate the effect upon management and productivity of the programming operations.

(6) A set of computer installation operating specifications designed to measure the effectiveness of the management of the installation from the keypoint of overall resource allocation. Resource allocation is defined as the use of equipment, manpower, and supplies.

Therefore, the total procedure is designed to measure the effectiveness of the performance of existing or projected computer installations. That is, given a computer system, software systems, programmers, operators, floor space, punch cards, reels of magnetic tapes, clerks, and librarians, how well does it or how well will it perform? The answers are given in a series of criteria defined in relative weights. These criteria represent standards by which any installation can be evaluated.

The question of installation performance is further complicated by consideration of what the installation is supposed to do, for example, since the same criteria do not all apply to a missile guidance system as to a personnel accounting function. Therefore we use an installation classification scheme to provide a basis for developing separate (probably overlapped) criteria and associated significance weights.

1.3.2 Objectives of the Evaluation Procedure

The AUERBACH Computer installation performance effectiveness evaluation procedure is designed to:

(1) Develop and furnish criteria to assist agencies in evaluating whether computers are being used effectively and in predicting the effectiveness of proposed computer installations.

(2) Expand existing policies for the selection of equipment to provide additional guidelines on:

(a) The preparation of system specifications to be transmitted to suppliers in requests for equipment proposals.

(b) Methods for evaluating suppliers' proposals.

(3) Develop and prescribe information necessary to provide selected managerial levels with information needed to effectively manage computer resources.

Lest the enthusiasm of the developers for what is believed to be a significant tool for USAF management be misconstrued as advocating a panacea for all problems associated with use, selection, and management of ADP installations, a note of caution is required. It is as important to understand what the process will not do as it is to understand its potential. For example, the system will not solve policy questions of any kind, it does not purport to answer directly whether an application should be automated, and it does not measure the performance of shared-memory multiprocessor systems.

This report will describe the details of equipment performance measurement and how it is combined with other factors that measure installation performance. The steps necessary to select from among competing proposals are also given, as well as a description of the self-improving and self-validating features of this approach. The concept of the extended machine, described in Section IV, is believed to be a significant major step forward in software performance evaluation. Furthermore, the derivation of a functional installation classification scheme permits the same installation to be measured on several types of functional assignments (as often found in both centralized service and integrated installations). Also, the use of estimated running time versus actual provides some measure of programming efficiency.

BLANK

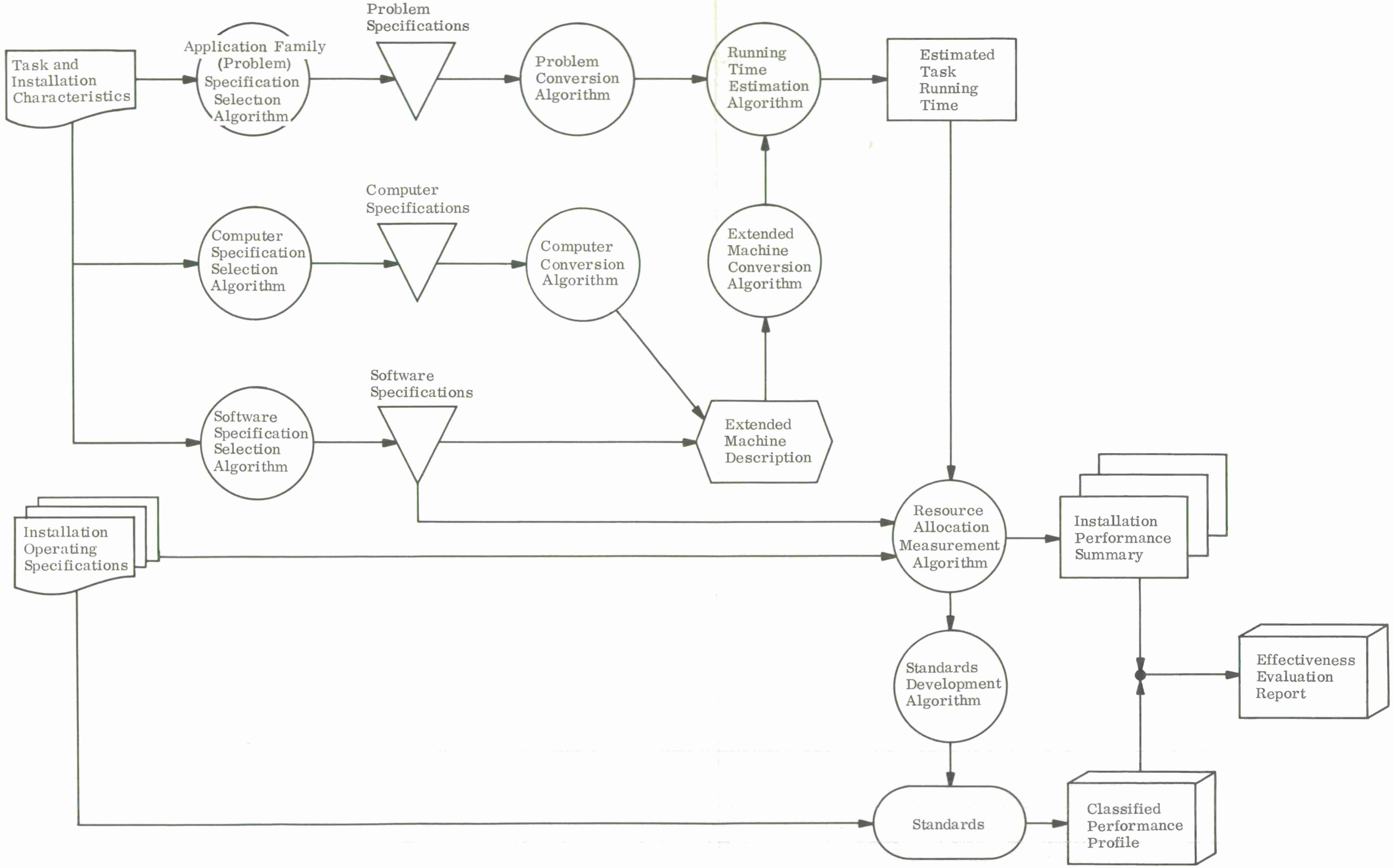

Figure 1. Procedure for Computer Installation Performance Effectiveness Evaluation

SECTION II

SUMMARY OF THE AUERBACH PERFORMANCE EVALUATION CONCEPT

2.1 THE AUERBACH PROCEDURE

The AUERBACH approach to computer installation performance effectiveness is based on the concept of utilizing quantitative data within an algorithmic process to produce varied measurements of effectiveness that are ultimately weighted to reflect their proper importance in determining overall computer installation effectiveness in terms of total cost. These independent sets of data can be gathered independently from separate groups and when processed through the evaluation procedure can be utilized to produce the measurements of effectiveness. The use of rigid definitions and standard procedures guarantees that the results of the performance evaluation or preinstallation justification will be objective. The overall procedure is illustrated in Figure 1.

2.2 COMPONENTS OF PERFORMANCE EVALUATION PROCEDURE

The independent sets of data are designed to measure the following areas.

2.2.1 Computer System Performance

Under previous work for the U.S. Air Force and through its efforts in Standard EDP Reports, AUERBACH has become firmly convinced that the only effective method to use in measuring computer performance is based upon a problem-oriented approach. In our previous contract we have demonstrated that it is possible to characterize the important hardware characteristics of a computer system in a series of macroloop functions, derive program execution times for these loops, and combine them with independently derived characteristics representing the significant features of a problem to yield a valid estimate of computer performance. In actual practice, it has been found that, since programming strategy and implementation of the strategy can vary over a wide range of performance times, the ranges of numbers produced for the performance of the macroloop functions will vary according to such constraints as limited central processor space, input-output speed limitations, central processor speed limitations, etc.

2.2.2 Hardware/Software Interaction - The Extended Machine

Recognizing that in most computer environments today a software language is used to write a majority of programs, we have chosen to measure the effect of the software upon the hardware performance independent of the problem because we feel it is the only way to apply universal measurements over a set of problem types. Therefore, we have developed what has been labelled an extended machine concept, whereby the interaction of software upon hardware is measured to produce some indication of the efficiency of the software and its effect upon performance. The extended machine performance measures are software-generated central processor performance times for selected macroloop functions. Efficiency is measured in terms of the aggregate ratios of previously derived estimated running times to those derived for the same functions under the software systems. This enables the procedure to use another algorithmic process to produce a modified or practical estimated running time using the central processor performance times generated by a given software system.

2.2.3 Problem Definition

Under the same contract it has been demonstrated that specific classes of problems, i.e., static file processing, dynamic file processing, on-line control, numeric data processing, etc., can be characterized by quantitative data representing systems requirements for a given problem. With this quantitative data and the quantitative data derived for the computer performance, it is then possible to use a rigidly defined algorithmic process to produce an estimate of computer running time. Provision has been made in the algorithm to account for the different types of computer structures, such as binary versus decimal, variable versus fixed word length, one address versus two address, etc.

2.2.4 Software Evaluation

The measure of software indicated in Paragraph 2.2.3 reflects its contribution to the computer system performance. Software also significantly contributes to or detracts from the utilization of programming talent. Therefore, in a qualitative evaluation, to be performed within rigid definitions and restrictions, we have developed a series of questions designed to measure the effectiveness of the software system in contributing to the effective utilization of programmers. A series of point scores or scalar values is derived for various features such as presence of diagnostics, source level or object level recompilation, documentation, etc., most of which are present to some degree in any software system. The total point score is then entered into an algorithm designed to produce a measure of the effective utilization of programmers as part of the resource allocation measurement.

2.2.5 Task Classification

The differences that exist among computer installations due to variations in operating objectives and operating requirements have complicated the understanding and management of computer installations.

A classification scheme is needed to reflect the different purposes for which computers are used, and the different operating requirements surrounding their use. Attempts to distinguish computer installations on the basis of types of equipment have been ruled out because the distinction between so-called scientific and business processing has faded with the maturation of computer technology. A classification by cost of equipment was eliminated because, as a general rule, the same problems of equipment utilization, staff utilization, and processing of work load face computers costing fifty thousand dollars to seventy-five thousand dollars as those costing one to two million dollars.

Our study indicated that the most logical distinction that could be made in classification of computer installations would be one that recognized the differences in environment in which the computer is operating, including the time within which the computer is required to provide a response. This throughput demand is in turn linked very closely to the collection of data from either local or remote sources. The service of these demands is affected by the environment for programmer operations - that is, the presence of an open-shop environment in support of a professional staff operation as opposed to central computing services (where professional programmers are used to supply computing services to varied types of demands) or the integrated operations environment (where known and similar types of application demands are served by professional programmers and operators).

2.2.6 Installation Specifications

Installation operating characteristics are not uniformly applied by managers of ADP installations, because of the lack of a common understanding as to what information to collect in order to measure the performance of an installation. Standard measurement criteria, if available, could be used by local management to evaluate the performance of its installation and to determine where improvements are needed. At department or other Government management levels such criteria would help in comparing proposed installation cost estimates against known costs for similar existing installations, pointing out problems needing corrective action and assessing total agency-wide effectiveness. In addition, the presence of performance criteria with related value guides would assist in the development of Government policies and guidelines and the evaluation of agency conformance to such policies.

2.3 EVOLUTION OF STANDARDS

Specific areas where guidelines or standards have been lacking include the following.

2.3.1 Number of Operating Personnel

It is our opinion that correlation of the number of operators, programmers, and supporting clerical staff can be developed as a function of the work load, response times, and computer environment. By collecting quantitative information relative to number and cost of operating and supervisory personnel, we hope to establish a correlation measure of people and costs necessary to meet given levels and types of demands within specific environments.

2.3.2 Operating Costs

Operations cost factors such as maintanance costs, mean time between equipment failures, monthly cost for supplies, and space charges should be correlated to provide relationships between throughput demands and installation environment. Thus it should be possible to estimate, for example, the costs of maintenance for integrated on-line operations as opposed to those for integrated batch processing.

2.3.3 Equipment Utilization

Statistics on equipment utilization concerning production time, total power-on time, and rerun time, correlated to number of production runs, program test runs, and compilations should be made available as guidelines for operating management. These guidelines must be restricted to particular computer environments derived from the classification scheme indicated above.

2.3.4 Schedule Performance

Installation performance should also be measured, on a statistical basis, in terms of the ability of the management of the installation to deliver completed projects on schedule. Source data must be collected to indicate numbers and variances in meeting projected delivery schedules. These figures again must be correlated to programmer, equipment, and operator availability within given environmental constraints.

2.3.5 Programmer Performance

Perhaps the most undefined area for the purposes of measurement is that of programmer performance. The collection of data in the area of programmer performance will undoubtedly impose certain record-keeping requirements upon installations not now collecting such data. We feel that the collection of labor hours data by work category (i.e., problem definition, flow charting, coding, check out, documentation, etc.) is necessary to bring about standards for measuring programmer achievement. Collection of this data under a standard procedure (as proposed in Section III) will add validity to the derivation of programmer work standards.

2.4 STATUS

The recent report to the President on the Management of Automatic Data Processing in the Federal Government, prepared by the Bureau of the Budget[3], submits that the derivation of standards for measuring performance is one of the most necessary aspects of the Government's ADP program. We submit that the procedure for the development of these standards must be based upon actual field operation of computer installations. The procedure designed to utilize this data should be self-improving, and for that reason we have chosen to incorporate a statistical technique known as stepwise multiple regression analysis. In this technique, the factors initially used as the data collection specifications for quantitative and qualitative data are entered into a statistical process during which those factors that prove to be insignificant in contributing to the overall installation cost will fall by the wayside. Relative weights of importance are then assigned to the remaining factors and their interrelationship with each other can be defined. From this type of analysis, numerical standards for individual factors can be established to be utilized as guidelines within varying installation environments.

In Section III of this report we have outlined our proposal for the development of the entire AUERBACH Computer Installation Performance Effectiveness model. The proposal requires collection of data according to the installation questionnaires illustrated in the appendices to this report and the development of the

---

[3] Report to the President on the Management of Automatic Data Processing in the Federal Government, Senate Document No. 15, 89th Congress, Bureau of the Budget, March, 1965.

algorithmic processes to interrelate each of these quantitative and qualitative specifications into an overall evaluation designed to determine the individual significance of each item within the independent data sets to be collected. The specifications indicated in the appendices are meant to be preliminary, but are indicative of the complexity of the task.

SECTION III

THE AUERBACH PROCESS IN ACTION

The expanding technology and an increased awareness of the potential of electronic data processing have increased the needs for further data automation. Historically, decisions regarding the adoption of a data processing system have been a costly compromise selected from many possible alternatives. It has been recognized by Air Force management that there is a need for guidelines by which they can make better decisions regarding future automation requests.

Not only has the Air Force management recognized the need for guidlines in making future decisions on data automation, but they have recognized the need for guidelines in measuring the effectiveness of existing installations. Both of these problems have been of major concern to the AUERBACH Corporation for many years. In fact, the perception of these needs gave rise to a continuing series of volumes entitled AUERBACH Standard EDP Reports, as well as the evolution of such tools as VECTOR and Technical Review and Critical Audit of installation performance. The process described herein is the extension of the concepts and tools that we have been developing over the last four to five years. Detailed explanation of the process to measure the effectiveness of existing and proposed installations is provided in Section IV. In this section, the actions necessary to implement the evaluation procedure are summarized.

The process described herein consists of two parts; the first part involves the development and initialization of the effectiveness evaluation procedure, and the second part is concerned with the use of the derived system by Air Force management. The first phase of the development of this system has been accomplished under our present contract with ESD. In this phase, we assigned a team of senior staff members to develop a concept to meet the need for establishing a set of criteria form which to measure computer installation performance. However, a concept in itself does not solve a problem. Therefore, we decided to go two steps beyond the conceptual stage of development: (1) we developed a methodology which extends our concept to the real world and (2) we selected a mechanism for testing our concept against realities of existing and proposed installations.

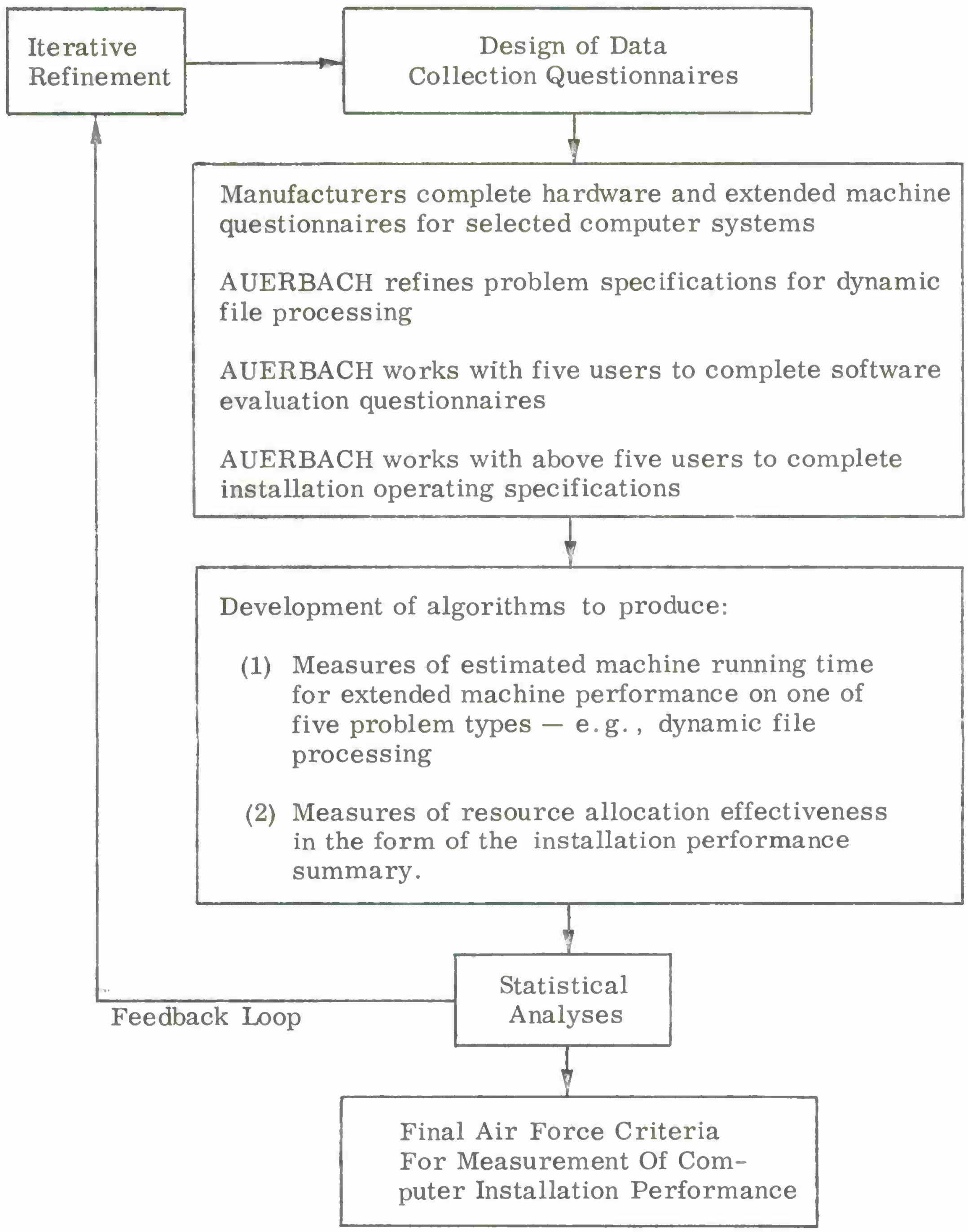

Figure 2. AUERBACH Plan for Dynamic Development of Computer Installation Performance Standards

Without the development of these latter two steps, we would be unable to validate our hypotheses regarding:

(1) How installations should be classified.

(2) What characteristics affect the effectiveness of existing and proposed computer installations?

In reference to (1) above, we developed a classification scheme. This scheme is derived from the functional classification of computer installation environment, and is based in part of the Bureau of the Budget Report.[4] It is not rigidly fixed, and it may be modified on the basis of the results of our data collection and analysis. This initial scheme classifies installations by 30 criteria, as discussed in Section IV. The classification scheme permits performance effectiveness comparisons of related problem families operating in separate and distinct environments. The interrelationship between these factors and other parts of the process is summarized through a series of conversion algorithms which are described in Paragraph 3.3.

Identification of the type of installation, problem class, and detailed specifications representing each component area is of little value without a definitive algorithmic process for combining them in such a manner as to arrive at a valid, objective measurement of performance effectiveness. The process developed by AUERBACH Corporation is a single tool for the selection of equipment and measurement of existing installations.

The overall AUERBACH process for measuring performance can be viewed structurally as iterative and self-refining. That is to say, it requires successive refinements of the data to arrive at the final effectiveness evaluation. The same logical operations are performed on the refined data in subsequent iterations of the process. It should be noted that the same logical sets of data are used in each iteration, but the preceding iterations help to select those data sets which require more refinements to arrive at the final measure. In other words, iteration is concerned with elements of installation and problem characteristics, and applicable hardware and software specifications, for example. It is this characteristic of the process which leads to its being regarded as self-refining.

It is also significant that the process can be viewed as self-correcting (see Figure 2). That is, subsequent applications of the process will, through multiple regression analysis and installation classification, lead to precise, refined identification of

---

(4) Ibid., pp. 9-12

TABLE I. PROPOSED DEVELOPMENT PLAN FOR COMPUTER INSTALLATION PERFORMANCE EFFECTIVENESS EVALUATION

ACTIVITIES \ MONTHS	1	2	3	4	5	6	7	8	9	10	11	12
1. Prepare Initial Questionnaire	—											
2. Plan Pilot Study	—											
3. Visit Pilot Installations		—										
4. Visit Computer Manufacturers		—	—									
5. Analyze Data			—									
6. Revise Questionnaire			—	—								
7. Develop Algorithms		—	—	—								
8. Collect Data From Installations				—	—	—						
9. Analyze Data (SMRA)						—	—	—				
10. Develop Standards								—	—			
11. Write User Guide									—	—		
12. Orient Air Force											—	
13. Write Final Report												—

important elements in performance effectiveness evaluation. Thus, the system will converge more rapidly toward a comprehensive, objective analysis with a minimal effort on the part of Air Force Management.

3.1 PROPOSED DEVELOPMENT PLAN

The procedure will be developed according to the plan shown in Figure 2 and the schedule shown in Table I.

At the outset, to evaluate the practical worth of the data collection questionnaires, a pilot study will be conducted in which five installations will be examined to:

(1) Refine the specifications and user guides.

(2) Determine the amount of time required to conduct each installation analysis.

After this initial stage is completed, the refined user guide and procedures for data collection will be completed for approximately 40 installations. AUERBACH's plan is to visit these installations in two phases to interview key personnel in order to complete the specification listed in the appendices. It is estimated that a two man team will be adequate to collect the appropriate data in two or three days at each installation.

From the data collected in the interview session, the installations will be classified into homogeneous groups based on the classification scheme. The validity of the classification scheme will, of course, be tested by statistical techniques, i.e., analysis of variance, etc. Then, the data collected for each group will be converted to summarized data elements derived through the algorithms and then keypunched on cards. The data on the cards will become inputs to the computer program for stepwise multiple regression analysis.

The multiple regression analysis will indicate which variables are significant and their relationships. The final product is used to develop a series of standards which will be documented along with the entire procedure in a users guide to Computer Installation Performance Effectiveness Evaluation.

Following this development, AUERBACH will orient key Air Force personnel with the aims and procedures utilized.

3.2 DATA COLLECTION

As indicated in Figure 1, there are two basic input questionnaires to be completed by personnel at various installations. These concern the data for scoring the software systems and the quantitative data on the administrative and financial aspects of the computer center operation. In addition, the one-page installation classification guide must be completed at each installation in order to determine the selection of representative problem specifications.

All of the other input data to be collected, i.e., computer specifications, extended-machine specifications, and problem specifications, would normally be drawn from the library of completed forms.

Eventually, data will be collected for all different types of installations and application classes. Initially, however, data collection will be limited to one type of installation (integrated operations), with sampling of a given number of those installations to test the methodology. Essentially, the approach consists of design of data collection questionnaries to gather information to be collected, collection of data through these questionnaires, conversion of the data to form composite data elements, analysis of the significance of these data elements by means of statistical techniques (i.e., multiple regression analysis), and development of a list of meaningful effectiveness evaluation criteria to be used by the Air Force.

It should be noted that this design approach is not static; rather, it is dynamic. This is accomplished by projected use of a feedback loop between the design stage and the analysis stage so that the system is continually being refined as a result of continued analysis. This can be seen in Figures 1 and 2.

The guides filled out by the manufacturer only once for each computer system and peripheral unit, i.e., computer specification, hardware/software (extended machine) specification, and software specification, are integrated with the problem specification (specified once, but completed each time to supply variable data on volumes and transaction activity) through the VECTOR process to obtain an estimated task running time, which is one input into the statistical analysis. (It is contemplated that this portion of the procedure could eventually be automated by developing a program to select the proper data entries and perform the mechanical calculations.) Additionally, the software specification provides a measure of the value of the software to the programmers and so becomes another input to the statistical analysis.

Installation environment characteristics, e.g., response time, complexity of tasks, and operating characteristics, number of programmers employed, number of analysts, for example, are also inputs to the statistical analysis. Furthermore, these statistics are used to provide installation management with an installation performance summary.

After the system is initialized, the Air Force need only collect data on the relevant variables from each type of installation because the variables may differ between installations. Periodically, the system can be tested to determine whether the installation characteristics have changed significantly.

The present intention is to collect computer and extended machine specifications for only eight to ten computer systems. The reasoning here is that such an effort would reduce the cost of implementation and still be sufficient to prove the validity of the procedure.

The total number of installations to be visited may approximate forty to fifty. These visits will be made in three phases: pilot, first group, and second group. In collecting installation-originated data, it may be possible to mail the problem, software evaluation, and installation operation specifications in advance of the actual visit in order to reduce the amount of effort spent on data collection. The reason for collecting the data in three phases stems from the belief that each round will provide a basis for changes in the questionnaries representing new items and deletion of non-significant items. The change cycle will be guided by the results of the multiple regression analysis and observations of installation operations. This is but one illustration of the dynamic nature of the process.

3.3 ACTION OF THE CONVERSION ALGORITHMS

There is a series of conversion algorithms that must be performed before the raw data collected via the questionnaires and specifications completed by manufacturers can be processed into the second by-product of the procedure — the installation performance summary. (The first by-product is a measure of software effectiveness produced by the extended machine ratios.) The summary report will represent a composite of such items as: estimated development time compared to actual development time, estimated completion time compared to actual completion time, production time compared to actual metered time, etc. (see Figure 3). It is intended that the summary be employed by installation management to improve performance.

INSTALLATION PERFORMANCE SUMMARY

I.	GENERAL	
	Installation Classification	________
	Principal application families by percentage workload	________%
II.	EQUIPMENT	
	Effective Machine Utilization	________
	Weighted average of each application family programming score	________
	Excess machine capacity	________
III.	SOFTWARE	
	Extended machine rating (software effectiveness)	________
	Evaluation score (software value)	________
	Percentage of total project (by application class) development time devoted to:	
	Source language coding	________%
	Program checkout	________%
	System test	________%
IV.	PERSONNEL	
	Percentage of total project development time devoted to:	
	Problem definition and analysis (computer oriented solution)	________%
	Source language coding	________%
	Program checkout	________%
	System test	________%
	Excess capacity	________%
V.	COST	
	Percentage of total installation cost represented by:	
	Computer system	________%
	Programmers/analysts	________%
	Operations personnel	________%
	Supplies	________%
VI.	INSTALLATION ACCOMPLISHMENT	
	Project development time rating	________
	Production system job completion time rating	________

Figure 3. Sample Installation Performance Summary

Another conversion algorithm is used to convert such raw computer specifications as start-stop time and recording density to an effective tape transfer rate, and time to process $A + B \longrightarrow C$, multiply execution time, simple update time, etc. into a composite. Similarly the extended machine performance elements are combined through another algorithm with the summarized data elements representing the problem specification to produce the estimated computer running time.

3.3.1 An Example of the Algorithm for Computing Estimated Running Time for Application Families Dynamic File Processing and Static File Processing.

Estimated running time is computed by using the Extended Machine Specifications (Appendix II of this report) and the Problem Specifications (Appendix III of this report). As an example, consider a file updating routine which requires (among other things):

(1) Some general input editing (alphanumeric).

(2) Both simple and complex updating macro-loops be performed.

(3) A table look-up.

(4) Some general commercial output editing (alphanumeric).

Part of the algorithm to determine application factors may be stated as follows:

The number of fixed alphanumeric input fields as computed as

$$\frac{(PS310) \times (PS340) \times [(PS320) - (PS330)]}{PS320} = \underline{\qquad\qquad} P1$$ [5]

The number of simple update steps per transaction record is computed as

$$(PS310) \times (PS360) = \underline{\qquad\qquad} P2$$

---

(5) The numbers PS nnn, ES nnn and IS nnn refer to specification numbers extracted from the appendices. In the algorithms, their appearance denotes the quantities which are the response to the specification. Numbers like P1, ER1, etc., are used as labels to denote quantities used in subsequent computations.

The number of complex update steps per transaction record is computed as

(PS310) x (PS370) = __________P3

The number of table reference steps per transaction record is computed as

(PS310) x [(PS360) + (PS370)] = _________P4.

The number of alphanumeric output fields per record is computed as

(PS710) x (PS740) = __________P5.

Part of the algorithm to determine the extended machine factors is shown below.

Time required for input editing task on alphanumeric input is computed as

ES110 = _____ E1

Time required to perform simple update is computed as

(ES101) + (ES105) = _____E2

Time required to perform complex update is computed as

(ES101) + (ES102) + (ES105) = ______E3

Time required for sequential table search is computed as

ES113 = _____E4

Time required for output editing task is computed as

ES111 = _____E5

Combining these values, the estimated running time to perform the program operations on a given machine using a given software system is derived.

Time required for input editing is computed as

P1 x E1 = _____ ER1

Time required for simple updating is computed as

P2 x E2 = _____ ER2

Time required for complex updating is computed as

P3 x E3 = _____ ER3

Time required for table look-up is computed as

$$P4 \times E4 = \_\_\_\_\_ \ ER4$$

Time required for output editing is computed as

$$P5 \times E5 = \_\_\_\_\_ \ ER5$$

Total estimated running time is computed as

$$(ER1) + (ER2) + (ER3) + (ER4) + (ER5) = \_\_\_\_\_\_\_\_\_\_$$

3.3.2 Algorithm to Complete Installation Summary Report.

Another example of how the algorithms are used can be seen in the production of the Installation Performance Summary. The Installation Performance Summary Report is divided into six sections. Section 1 contains general information relating to the installation classification and principal application classes. Thus, the portion of the algorithm used to complete this section could be stated as follows:

> The installation classification is presented in the installation and problem classification matrix. Enter this information in the Summary Report.
>
> The principal activity classes for the installation have been derived formally through a decision table using the classification matrix. The percentage workload for each class is stated in Installation Operating Specification IS402. Enter this figure in the summary report.

Section 2 summarized computer equipment utilization.

> To determine the percentage of effective machine utilization, compute the ratio of metered time to power-on time.
>
> $\frac{IS302}{IS301} = \_\_\_\_\%$, percentage of effective machine utilization.
>
> The weighted average of each application class programming effectiveness score is determined by the ratio of estimated running time for each application class to the actual running time, considering the percentage of total installation workload represented by the application class as the weight.

$$\frac{IS407}{IS408} \times (IS042) = \_\_\_\_\_ \text{ weighted application class programming score}$$

Excess machine capacity is defined as the difference between 176 hours (or whatever other number is selected) and the sum of maintenance time and metered time.

176 - (IS201 + IS301) = Excess machine capacity

Section 3 of the Installation Summary Report is an analysis of the software efficiency in terms of hardware/software interaction, software evaluation based on features of the system, and the project time devoted to use of the software system.

The extended machine rating is defined as the ratio of the sum of estimated running times for selected macroloop functions as performed in the hardware to the sum of estimated running times for the same macroloops as coded in the software system.

$$\frac{(CS201)+(CS202)+(CS203)+(CS206)+(CS207)+(CS208)+(CS212)+(CS311)+(CS312)+(CS325)+(CS326)+(CS327)+(CS217)+(CS218)}{(ES101)+(ES102)+(ES103)+(ES104)+(ES105)+(ES106)+(ES107)+(ES108)+(ES109)+(ES110)+(ES111)+(ES112)+(ES113)+(ES114)+(ES115)+(ES116)} = \_\_\_\_\_$$

The software evaluation score is the total score of the software specification. Sum the individual scores entered in the Software Specification Form as indicated and enter this figure in the Installation Summary Report.

Percentage of total project time devoted to utilization of software system is computed as

$$\frac{(IS504)+(IS505)+(IS506)+(IS507)}{IS406} = \_\_\_\_\_\%$$

Allocation of personnel resources includes problem definition time as well as programming and system integration time. This summary is requested on a per task basis. In personnel allocation as well as in machine utilization, excess capacity is a useful measure of effectiveness. These summaries are derived below.

Percentage of total project development time utilized by installation personnel is computed as

$$\frac{(IS501)+(IS502)+(IS503)+(IS504)+(IS505)+(IS506)+(IS507)}{IS406} = \_\_\_\_\_\%$$

Excess capacity is determined as the difference between 176 hours/man/month and the time actually spent on all tasks by programming personnel. Thus, for <u>all</u> application classes

Sum of (IS501) - (IS509) = _____

Enter: IS111_____
IS113 ____
IS117 ____
IS119 ____

[(IS111)+(IS113)+(IS117)+(IS119)] x 176 - Sum of IS501-IS509

(as entered above) = _____ Excess programming capacity.

Part 5, cost summary data is concerned with the percentage of total installation cost represented by the computer, personnel, and supplies.

Total Cost (TC) = (IS102)+(IS104)+(IS105)+(IS107)+(IS108)+
(IS112)+(IS114)+(IS115)+(IS116)+(IS118)+
(IS120)+(IS112)+(IS206)+(IS207)+(IS209)+
(IS211)+(IS213)+(IS215)+(IS304)
= ______(TC)

Percentage of total cost allocated to computer equipment is computed as

$$\frac{\text{IS304}}{\text{TC}} = \_\_\_\_\_\%$$

Percentage of total cost allocated to programmers and analysts is computed as

$$\frac{[(\text{IS112})+(\text{IS114})+(\text{IS115})+(\text{IS116})+(\text{IS118})+(\text{IS120})]}{\text{TC}} = \_\_\_\_\_\%$$

Percentage of total cost allocated to operations personnel is computed as

$$\frac{[(\text{IS102})+(\text{IS104})+(\text{IS105})+(\text{IS107})+(\text{IS108})]}{\text{TC}} = \_\_\_\_\_\%$$

Percentage of total cost allocated to supplies is computed as

$$\frac{[(\text{IS206})+(\text{IS207})+(\text{IS209})+(\text{IS211})+(\text{IS213})+(\text{IS215})]}{\text{TC}} = \_\_\_\_\_\%$$

In Part 6, Installation Accomplishment, ratings are determined which measure some estimated versus actual delivery times.

The project development time rating is to be computed on a per task basis. It is defined as the ratio of estimated development time to actual development time. Compute as

$$\frac{IS403}{IS404} = \_\_\_\_\_\_ \text{ project development rating}$$

Job completion time rating is to be computed on a per task basis. It is defined as the ratio of estimated completion time for a production system to actual completion time for production system. The final ratio will be stated as + or - dependent on how well response time requirements are fulfilled. The requirements are stated in the installation and task classification matrix. Compute as

J1. $\frac{IS405}{IS406} = \_\_\_\_\_\_$ job completion rating ratio

J2. Enter "+" if IS406 falls within the response time requirements stated in installation and task classification matrix.

Enter "-" if IS406 does not fall within the response time requirements stated in the installation and task classification matrix.

Job completion rating is entered as the juxtaposition of the values of J1 and J2.

3.4 OUTPUTS

The summarized data elements resultant from the algorithms in regard to the variables, e.g., number of analysts, number of programmers, complexity of task, programmer effectiveness, equipment utilization effectiveness, etc., will be analyzed by means of the stepwise multiple regression technique. The output of such an analysis would be:

(1) The degree of relationship between the significant variables (and their relative weights) and dollars expended. If the degree of relationship is statistically significant, then the significant variables

and their relative weights do indeed explain the dollar variation. If this is not the case, then more data must be collected from the installation. Thus, there is available a mechanism for validating the hypothesized variables.

(2) A standard for each installation based on the variables uncovered as significant and the relative importance of each variable will also be calculated. Furthermore, the analysis will compare the actual dollars expended to the calculated standard dollars to ascertain the degree of deviation of the actual from the standard to give a true measure of performance effectiveness in terms of dollars expended for work achieved.

Since the relationship of the significant variables and the dollars expended is expressed in terms of an equation, the sensitivity of each variable can be tested by means of a sensitivity analysis to study the impact of the variables on the dollars expended. As a result, the relative impact of the variables commensurate with their values can be ascertained.

In the second stage, the derived equation can be employed by the Air Firce, to provide a score for an existing or proposed installation. The score for the installation is computed in the following manner. The initialization program provides a set of variables to be measured. In addition, it provides a numerical weight for each variable. Thus, the Air Force has a set of variables and their associated weights. For a given installation, the Air Force collects data for the relevant variables. The values obtained are multiplied by the weights to obtain scores for those variables. By summating the values for all of the relevant variables, the expected dollars expended is calculated. Comparing expected dollars to actual or budgeted dollars gives a measure of effectiveness for that installation.

Similarly, the same process would be used in evaluating new proposals. The only difference being is that estimated dollars would be compared to expected or predicted dollars. Then the performance index is calculated by the following formula:

$$\frac{\text{Actual \$}}{\text{Standard \$}}$$

Periodically, the system is tested to insure that the values and weights have not changed significantly. This test is performed by means of the Stepwise Multiple Regression Program.

In the final step of the iteratively refined process, the Air Force will be provided with a set of significant criteria and their relative weights for a given installation class in the form of the performance profile seen in Figure 4. Note that there is a different profile format for each installation type.

Data derived from the multiple regression analysis is entered on the form in the appropriate column. The value is then multiplied by the weight to obtain the total score for that variable. The standard total score for each variable is simulated to obtain the standard dollars, which is then compared to the actual dollars expended.

3.4.1 By-Product Outputs

Perhaps the most significant by-product of the entire process is the establishment of uniform procedures for recording of measurement data concerning the allocation of data processing resources. In this light, the presentation of the Installation Performance Summary (Figure 3) provides installation management with a measurement of its effectiveness in allocating and controlling the resources of computer and supporting equipment, personnel (including managers, analysts, programmers, data preparation clerks, and control clerks) and supplies.

An important by-product of the procedure is obtained from the production of an estimated computer running time. This estimate is used as a common denominator or de facto standard for comparison against the actual running time. It provides a measurement of the effectiveness of the programmers in preparing working programs using a given software system for a given machine configuration.

The development of an extended machine concept provides a measurement of the effectiveness of a given software system on a particular machine configuration. In effect, this is a measure of how well the software can be expected to perform on a given computer system configuration.

A valuable by-product of the computer performance figures is an indication to the prospective user of potential problem areas to be encountered for each specific computer system in preparation of the key program runs. Since the performance

Significant Characteristics	Installation Class ________ Application Family ________ Criteria	Relative Weight	Predicted Dollars	Standard Dollars
No. of Programmers	1. ________	______	______	
No. of Analysts	2. ________	______	______	
Equipment Costs	3. •	•		
Programming Efficiency	4. •	•		
Software Efficiency	5. •	•		
Software Value	6. •	•		
	7. •	•		
	Standard Dollars			______
	Actual Dollars		______	
	Performance Index		______	

Figure 4. Sample Installation Performance Profile

figures are derived separately under various programming strategies (i.e., central processor speed limited, input-output speed limited, core storage space limited, etc.), the programming manager has a useful guide to what problems may be inherent in a particular approach.

The system analyst, by following the standard procedure used to describe the application parameters in Appendix III, gains a better appreciation of the important parameters to be gathered in a system analysis study. Furthermore, the use of the problem specifications on a uniform basis throughout a range of Air Force installations will permit a more meaningful statement of system requirements (as recommended by the Bureau of the Budget Report).[6]

---

[6] Ibid, Chap. 7, pp. 47-51.

SECTION IV

STRUCTURE OF THE AUERBACH PROCEDURE FOR COMPUTER INSTALLATION EFFECTIVENESS EVALUATION

The overall structure of the AUERBACH procedure is shown in flow chart form in Figure 1 of this report. In general, the process may be described as involving:

(1) The collection of data regarding the computer installation environment and its utilization of resources.

(2) The combining of this data with available data measuring the significant performance factors of the computer system, software system interaction, and problem requirements, to yield an installation performance summary.

(3) The statistical treatment of this data to provide measurements of computer installation effectiveness and standards against which the effectiveness values for installations may be compared.

4.1 GENERAL DESCRIPTION OF THE PROCESS

The independently derived data sets used in the AUERBACH process are a set of specifications and characteristics regarding the principal tasks of an installation, its computer complex, and administrative and financial data describing the installation. The data is combined algorithmically to produce estimated running time for the computer complex as applied to a well-defined problem which is representative of the task classification. This, however, is only one element of the process, and in effect gives an intermediate result which helps Air Force Management determine how effectively the computer equipment is being utilized with respect to programming efficiency. The data which is collected concerning the operating characteristics of the installation is treated statistically to determine the effective allocation of dollar resources.

The intermediate results indicated above are of direct use to middle management, e.g., the computer installation manager, in the evaluation of his local situation. For higher management, at the command level or above, this information is further refined to provide an effectiveness profile of the installation. This profile relates the dollar allocation to the efficiency of the task performance, dollars being used as a common denominator in the evaluation.

The AUERBACH process, since it provides objective measures of performance efficiency based on both quantitative and qualitative data, also provides standards for Air Force management in determining installation effectiveness. This is accomplished in part through standard, objective specifications provided in the design of the data collection questionnaires. The objectivity and standardization of this data are also enhanced by the very structure of the process, which is algorithmic in nature, and hence guarantees that a measure of effectiveness can be provided for any general-purpose[7] computer installation. The specifications and characteristics have been divided into four nearly autonomous categories: computer hardware, extended machine (hardware/software interaction), software evaluation, and problem specification. These specifications are completed once and once only, within the framework of the task and installation classification, either by Air Force personnel, the computer manufacturer, or by independent sources, depending on the nature of the specification and Air Force requirements. Since the specifications are standardized, and are combined algorithmically by well-defined rules, the AUERBACH process, once developed, will become a low-cost management tool for the Air Force. It can be used for evaluation of the effectiveness of existing installations and objective evaluation of proposals for new installations by use of the guidelines for dollar cost as related to each criterion and its empirically derived standard.

4.2 INSTALLATION AND TASK CLASSIFICATION

The above general description of the AUERBACH process refers to a task and installation classification scheme which is the unifying "force" of the process. This scheme is very similar to the Bureau of the Budget classification as shown in Report to the President on the Management of Automatic Data Processing in the Federal Government[8]. Superimposed upon the Bureau of the Budget scheme is a task classification procedure. The approach adopted in our study prior to publication of the Bureau of the Budget report was so similar to that of the report that the format and terminology of the latter were adopted to avoid confusion.

[7] While the process is generally applicable for special-purpose installations, such as SAGE, these have been excluded from our installation classification at the present time.

[8] Ibid., pp 10-12.

		PROFESSIONAL SUPPORT	CENTRAL COMPUTING SERVICES	INTEGRATED OPERATIONS
Principal Activity				
Response Time Requirements	All Time Scheduled			
	Time Critical; Greater Than 1 Hour			
	Time Critical; Greater Than 5 Minutes But Less Than 1 Hour			
	Time Critical; Less Than 5 Minutes			
	Time Critical; Less Than 30 Seconds			
Source and Mode of Receipt of Raw Input Data	Local & Batched			
	Local & Random			
	Remote & Batched			
	Remote & Random			

Figure 5. Installation and Problem Classification Matrix

Our classification scheme, then, is complementary to that presented by the Government. It says, in essence, that not only is an installation classified by its environment, but also by the task(s) which it is assigned to perform.

The rows of the AUERBACH classification table are defined by three major specifications: principal activity, response time requirements, and source and mode of receipt of raw input data. The three major divisions in turn are delineated by more detailed specifications. For example, under response time, we delineate the rows as scheduled operations, response time requirements greater than one hour, etc. Under source and mode of receipt of raw input data, we inquire as to whether the source of the data is remote or local, and whether the mode of input is batched or random.

In the course of this project, five generic problem classes which represent many important computer applications have been identified and are defined below. The five classes are:

(1) Dynamic File Processing

(2) Static File Processing

(3) Numeric Computation

(4) Non-numeric Processing

(5) On-line (Process) Control

These problem classes are, in effect, families of applications. However, the generic names applied here allow us to classify computer installation environment by task as the task is accomplished in a particular environment. This is very important in developing standards for purposes of effectiveness evaluation. It is possible to make comparisons based on a whole range of specific subtasks of an installation, rather than one isolated task. Furthermore, rather than compare installations on the basis of inventory control applications for example, which are implemented with widely varying methods of processing, equipment, and data requirements, installations are compared which have inventory control applications accomplished through either dynamic file processing or static file processing. Thus, similar types of application design within specific installation environments are compared against each other.

Definitions of the five families of problem classes and indications of their derivation from the classification scheme follow:

(1) Dynamic File Processing is file processing in which input transactions may be processed randomly with respect to stored data records that are either randomly stored or linked together via an accession sequence. It is also characterized by noting that a single item of data or a set of data is transformed in the processing so that the output can appear in more than one distinguishable form. Processing may be scheduled or nonscheduled with response time requirements measured in seconds or minutes.

(2) Static File Processing is file processing in which the records are stored sequentially with respect to the input data. Processing usually includes the updating of a record and its output in a unique form. Processing is generally scheduled, with response time requirements measured in hours or days.

(3) Numeric Computation is the processing of and computation with numeric data, which is often characterized by a large series of iterative operation loops and high precision, and utilizes such mathematical/statistical techniques as matrix inversion, regression analysis, linear programming, etc. Results of computation generally provide numerical values as the output. Processing is generally scheduled within twenty-four hour periods after receipt of data and programs. If nonscheduled, response time requirements can be expected to be in minutes.

(4) Non-numeric Processing is processing of data which includes primarily alphanumeric messages and raw data numbers. It is frequently characterized by relatively small permanent files as compared to larger files of intermediate data storage, and an exceptionally high incidence of decision making and branching operations. It often includes processing of data which is truth-functional (true/false, yes/no) rather than numerically quantitative. Prime examples of this type of processing can be seen in simulation and other modeling applications, and in heuristic programming. Processing may be scheduled or nonscheduled with response time requirements measured in minutes or hours.

(5) On-line (Process) Control is the control of continuous process operations within a real-time environment. That is, output of the system is used to initiate actions that will be processed to provide subsequent feedback or input. Processing is generally nonscheduled, with response time requirements measured in seconds or minutes.

Figure 6 illustrates via a composite diagram the use of the AUERBACH classification scheme. Note that the composite matrix takes the form of a decision table. The five application families are noted in the matrix by the number associated with each name. In practice, the composite matrix will not be used, each requirement being checked (√) where applicable. Thus, the application family is determined by the incidence of checks in each column, and the classification scheme itself is formalized through decision table techniques. In Figure 6, however, the number is used in place of a check (√) in the matrix to afford easy discrimination between the application families and their associated characteristics.

Referring to Figure 6, the illustrative task and installation classification matrix, the stated applications are:

(1) Engineering design

(2) Research and development

(3) Inventory control

(4) Information storage and retrieval

(5) Regression analysis

(6) Dynamic simulation

(7) Missile guidance.

The family called <u>dynamic file processing</u> is easily distinguished through the response time requirements and the fact that input data is characterized as random. Inventory control is an example of an application which can fit into this family. Whenever the dynamic file processing application family is selected in the decision table, the installation is classified as one in which this family plays a dominant role.

The other families are derived in the same manner. <u>Static file processing</u> can usually be found in either Central Computing Services or Integrated Operations. However, the raw input data would be batched, response times are in minutes or hours, and processing is scheduled. It is seen through these specifications that the Static File Processing family is precluded by the characteristics given for dynamic file processing. In the column headed "Central Computing Services," the characteristics might lead to some confusion between static file processing and numeric computation. However, the term "file processing" in the name and the definition of the

		PROFESSIONAL SUPPORT	CENTRAL COMPUTING SERVICES	INTEGRATED OPERATIONS
Principal Activity	(Sample Application Entries)	Engineering Design and Research and Development	Inventory Control, Information Storage and Retrieval, Regression Analysis	Inventory Control, Information Storage and Retrieval, Dynamic Simulation, Missile Guidance
Response Time Requirements	All Time Scheduled	3	2, 3, 4	2
	Time Critical; Greater Than 1 Hour	3	2, 3, 4	2, 4
	Time Critical; Greater than 5 Minutes But Less than 1 Hour		1	1, 4
	Time Critical; Less than 5 Minutes			
	Time Critical; Less than 30 seconds			5
Source and Mode of Receipt of Raw Input Data	Local & Batched	3	2, 3	2, 4
	Local & Random		1, 4	1, 5
	Remote & Batched	3	2, 3	2, 4
	Remote & Random		1, 4	1, 5
	Dynamic File Processing (1)		✓	✓
	Static File Processing (2)		✓	✓
	Numeric Computation (3)	✓	✓	
	Non-Numeric Processing (4)		✓	✓
	On-line (Process) Control (5)			✓

Figure 6 - Composite Installation and Problem Classification Matrix

families assume that the processing will utilize prestored records on files. Hence, the distinction can be made on the basis of name.

The numeric computation family generally occurs in the Professional Services or Central Computing Services categories. Engineering and scientific applications are usually included in this family. The input is generally batched, and response time requirements would be noted as being scheduled or the limitations would be less severe than, for example, seconds.

Non-numeric processing could occur in all three installation categories, but is highly unlikely in Professional Support. In Figure 6, the random input would preclude selection of this family, as would the type of application. Since Simulation is a prime example of non-numeric processing, applications with similar problem statements are likely to fall into this family.

Process (on-line) control probably would not be found in the Central Computing Service category or in the Professional Support category. The predominant characteristics of this family will be found in the priority response time requirements of the matrix.

The above analysis indicates how the AUERBACH task and installation classification method provides a concise, well-defined process to classify a computer installation by task as well as by environment Since we have adopted the Bureau of the Budget names for the types of installations, we have also retained their definitions of the various installation types.

A set of representative problem specifications is also selected at the same time the installation classification is accomplished. Details of these problem specifications will be presented in Paragraph 4.3.3.

In the effectiveness evaluation of an existing installation, the computer system being utilized is known. In the event that equipment is to be selected,[9] the same task and installation classification scheme used above can provide guide lines for the selection. We have, for example, adopted the Bureau of the Budget characterization

---

[9] The AUERBACH process for effectiveness evaluation considers the selection and evaluation of existing installations to be logically identical. This is due to the fact that the same questions must be answered in either case, the only difference being the times at which the questions are asked.

of an installation classified as Integrated Operations. Hence, when this class is shown to exist through the task specifications, the selection process should start with medium- to large-scale equipment. The source and mode of raw data input indicates whether remote input devices and communication equipment are required. Furthermore, the mode can be a useful indicator as to the need for random access devices. In the same sense, response time requirements can be useful in determining the required speed of the computer system.

4.3 SPECIFICATIONS

For expository purposes, assume that a specific computer system has been selected for evaluation. It is now possible to examine in more detail the specifications as shown in the Appendices of this report.

The specifications contained in Appendix I are designed to reflect accurately the details of a computer system.

For the current project, specifications have been included for line printers, card equipment, and random access devices, as well as some for the central processor as a prelude to including multiprogramming operations and multiprocessor systems in the analysis. It is important to note, however, that the VECTOR process actually comprises a relatively small part of the entire computer installation effectiveness evaluation procedure. It should be noted that for use in the effectiveness evaluation process, the computer specifications need be completed only once for a particular computer system. The specifications are divided into several parts, such as processor times, input/output times, magnetic tape specifications, etc. The specifications are stated in such a manner that the response is useful in computing the estimated run time for a particular problem. Raw add times as stated in a manufacturer's advertising brochure, for example, are not of primary concern.

The specifications do, however, take into account the time required to add two four-digit operands in main memory and store the sum in main memory. Thus, the specifications reflect the system as it will be used in an application, rather than raw times, which are generally meaningless by themselves. The specifications are quite detailed, and it is suggested that the response be completed by a technical representative of the manufacturer, in order to reflect the true characteristics of the machine. Since these specifications represent a standardized method of cataloging computer characteristics, systems presented by each manufacturer are viewed without bias. Hence, it is reasonable to ask the manufacturer to complete the specifications as a condition of doing business with the USAF. It is estimated that a technically competent manufacturer's representative can complete these specification forms in approximately three man-days.

4.3.1 Computer Specifications

The computer specifications have been divided into several parts, as listed below:

Part 1 - System Specifications
Part 2 - Processor Time
Part 3 - Input/Output Times
Part 4 - Magnetic Tape Specifications
Part 5 - Line Printers
Part 6 - Card Equipment
Part 7 - Random Access Devices

In Part 1, specifications are derived for such items as the size of main memory, word size, etc. In addition, specifications pertinent to the number of input/output data channels, line printers, and processors, for example, are included. These specifications are straightforward and easily completed by the manufacturer. Examples of these specifications are shown in Figure 7.

Part 2, Processor Time, is also straightforward. Care must be exercised in completion of these specifications, however, to insure accurate response. The specifications are, as the title implies, processor times. The time required to perform arithmetic operations in main memory and store the result in main memory, for example, is of interest. Raw add times of an arithmetic unit represent only one factor in determining the times; access times, addressing schemes, etc., also enter into these specifications. In addition, timing information is requested with respect to certain executive functions, such as time required to respond to an I/O interrupt condition. While these specifications are generally derived with respect to the hardware/software interaction of an operating system and a machine, many of these features are built into the hardware, and these times are collected in this section. Some of these specifications are illustrated in Figure 8.

In Part 3, Input/Output Times, the specifications detail most of the functions associated with input/output operations. Most of the possible permutations of these functions, such as verification of card images, and pre-editing of line images

COMPUTER SPECIFICATIONS

PART 1 - SYSTEM SPECIFICATIONS

SPECIFICATION NO.	APPLICABLE FOR					SPECIFICATION	RESPONSE
	B W	D W	D C	A W	A C		
CS 101	x	x		x		Main memory size in words.	words
CS 102		x		x		Word size in alphanumeric characters.	chars.
CS 103		x		x		Word size in decimal digits.	digits
CS 104	x					Word size in bits (excluding check bits).	bits
CS 105					x	Main memory size in characters.	chars.
CS 106			x			Main memory size in decimal digits.	digits
CS 107			x			No. of decimal digits per alphanumeric character.	
CS 108	x	x	x	x	x	No. of index registers.	
CS 112						No. of input/output data channels.	
CS 113						No. of line printers.	
CS 114						No. of card readers.	
CS 115						No. of card punches.	
CS 116						No. of random access devices.	
CS 117						No. of processors.	

Figure 7. Sample of System Specifications

COMPUTER SPECIFICATIONS

PART 2 - PROCESSOR TIMES

SPECIFICATION NO.	APPLICABLE FOR BW	DW	DC	AW	AC	SPECIFICATION	RESPONSE
CS 201	x	x	x	x	x	Time taken to add two operands in main memory and store the sum (operands must have more than four decimal digits).	μsec.
CS 202		x	x	x	x	Time taken to multiply an X digit operand by a Y digit operand and store the product (X and Y must be greater than 4).	μsec.
CS 203		x	x	x	x	Time taken to divide an X digit operand by a Y digit operand and store the quotient (X and Y must be greater than 4).	μsec.
CS 204	x					Time taken to multiply two operands in main memory and store the product.	μsec.
CS 205	x					Time taken to divide two operands in main memory and store the quotient.	μsec.
CS 206	x	x	x	x	x	Time taken to index in operand.	μsec.
CS 207	x	x	x	x	x	Time taken to compare two operands in main memory (of at least eight decimal digits or equivalent) and to transfer control to one of two arbitrary locations based on the result of the comparison.	
CS 221						Time required to respond to hardware interrupt condition not due to hardware malfunction, and transfer control to program execution mode.	μsec.
CS 222						Time required to respond to interrupt due to hardware malfunction, take whatever corrective action is possible, and transfer control to program execution.	μsec.
CS 223						Time required to respond to priority job interrupt conditions and transfer control to program execution.	μsec.
CS 224						Time required to transfer control to alternate hardware processor.	μsec.
CS 225						Main memory requirements for resident operating system.	words/char.

Figure 8. Sample Processor Times Specifications

to be printed are included. The times requested in this section are mostly related to the internal processor editing functions, and are only incidentally related to the speed of the input/output device, e.g., line printers and card equipment. The latter specifications are detailed separately in Parts 5 and 6, respectively, of the Computer Specifications.

In asking the manufacturer to complete Part 3, it is important to stress the facts stated above. It should also be noted that these times are particularly important in deriving accurate estimates of running time (computer system performance), especially when the representative problems of dynamic file processing and static file processing are the principal application classes to be considered. In these specifications most of the permutations an analyst might consider are noted. Hence if the responses are made accurately, the computer system will be shown without bias in this aspect of the analysis. Figure 9 illustrates a few of these specifications.

Part 4 of the Computer Specifications is a set of detailed specifications designed to cover all operational aspects of magnetic tape subsystems. It is illustrated in Figure 10. As in the other parts, the specifications take into account more than "advertising type" transfer rates. Rather, the tape subsystem is shown as it would be used when integrated into an operating computer system. There are specifications which deal with hardware performance timings concerning effective use of the tape units as well as specifications which deal with the manner in which the tape subsystem will affect the central processor available time. While these specifications are, for the most part, very straightforward, it is important that they be completed as accurately as possible, in order to reflect the true operational capability of the magnetic tape subsystem.

Line Printer specifications are covered in Part 5 of the Computer Specifications. Accurately completed, these specifications provide a true reflection of the line printer under consideration. It is important to note that certain of these specifications call for "effective printer speed" under a given set of conditions. Effective speed is based upon the time the printer is actually in use for the purpose of printing a single line, including variable factors such as start-stop times. Since the entire set of computer specifications will probably be completed by the manufacturer, it is felt that these times can best be determined by experimentation with the printer. Hence, empirically derived quantitative results as the response to these specifications are expected. Figure 11 is an example of the line printer specification sheet.

COMPUTER SPECIFICATIONS

PART 3 - INPUT/OUTPUT TIMES

SPECIFICATION NO.	APPLICABLE FOR B W	D W	D C	A W	A C	SPECIFICATION	RESPONSE
CS 301	x	x		x		General input editing task* with programming minimized and 11-character alphabetic field. Input field is synchronized (i.e., aligned in accordance with computer word structure).	μsec.
CS 302							
CS 308	x	x		x		General input editing task* with object time minimized and 5-digit numeric field. Input field is not synchronized.	μsec.
CS 309			x		x	General input editing task* with programming minimized and 11-character alphabetic field.	μsec.
CS 310			x		x	General input editing task* with programming minimized and 5-digit numeric field.	μsec.
CS 315	x	x		x		General output editing task** with programming minimized and a 6-character numeric field and scientific editing. Output field is synchronized.	μsec.
CS 316	x	x		x		General output editing task** with object time minimized and an 11-character alphabetic field. Output field is synchronized.	μsec.
CS 325			x		x	General output editing task** with programming minimized and an 11-character alphabetic field.	μsec.
CS 326			x		x	General output editing task** with programming minimized and a 6-character numeric field and commercial editing.	μsec.

Figure 9. Sample Input/Output Timing Specifications

COMPUTER SPECIFICATIONS

PART 4 - MAGNETIC TAPE

SPECIFI-CATION NO.	SPECIFICATION	RESPONSE
CS 401	Number of magnetic tapes which can be reading with processing proceeding.	
CS 402	Number of magnetic tapes which can be reading with no processing proceeding.	
CS 403	Number of magnetic tapes which can be writing with processing proceeding.	
CS 404	Number of magnetic tapes which can be writing with no processing proceeding.	
CS 414	Number of decimal digits per alphanumeric character in the computer's internal code.	
CS 415	Number of decimal digits per alphanumeric character in the magnetic tape code.	
CS 416	Number of alphanumeric characters per computer word.	
CS 417	Maximum blocking factor for card image input available using standard routines.	
CS 418	Maximum blocking factor for line images output available using standard routines.	

Figure 10. Sample of Magnetic Tape Subsystem Specifications

COMPUTER SPECIFICATIONS

PART 5 - LINE PRINTERS

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
CS 501	Skip speed(s)	inches/sec.
CS 502	Effective printer speed* for alphanumeric character set (letters A-Z; numerals 0-9, and 4 special symbols) at interline space:	
	1/2 inch	lpm
	1 inch	lpm
	2 inches	lpm
	3 inches	lpm
	4 inches	lpm
	5 inches	lpm
	6 inches	lpm
CS 503	Effective printer speed for numeric character set (numerals 0-9 and 4 special symbols) at interline space:	
	1/2 inch	lpm
	1 inch	lpm
	2 inches	lpm
	3 inches	lpm
	4 inches	lpm
	5 inches	lpm
	6 inches	lpm
CS 504	Print width of printed page	char

Figure 11. Sample of Line Printer Specifications

In Part 6, specifications are presented for punched card equipment, i.e., card readers and punches. As in the case of line printers, effective speed under certain conditions is of concern, and responses are expected to be derived empirically. These specifications also cover both clutch operated and non-clutch operated card equipment, so that nearly all presently used makes of card equipment can be included. Since punched cards play an extremely vital role as a major input medium in many Air Force applications, these specifications can significantly influence the anticipated running time of a computer system. Therefore, it is important to stress the accuracy of these specifications. In Figure 12, a representative example of the card equipment specification form is shown for both the reader and punch.

Detailed specifications for random access devices, Part 7 of the Computer Specifications, comprise the longest single part of the specifications. This is due to the fact that the responses are, in most cases, table entries which cover effective speed of the device and channel usage times for various file and record sizes. It is especially important to note that these specifications cannot be completed from raw hardware data alone. The person responsible for this set of specifications must know something about the file and record sizes where the random access device is to be applied.

For each specification CS701-708, there are several conditions included in the specification. These conditions are to be considered as being joined, i.e., "anded" together. We have considered in the specifications two distinct ways of using random access devices, namely, maximizing the number of records which can be processed in a unit of time or, alternatively, minimizing access time to a given random record. The specifications should be completed for each case as noted. The application analyst independently determines which case is required in his application, and the problem representing that case will be used as one of the problems testing programming effectiveness.

There are also two different types of random access devices which are included in these specifications. For convenience, we call them nonremovable, typified by fixed, revolving disc devices, and removable, typified by cartridge devices (i.e., CRAM, DATA CELL, RCA 3488). In the case of the latter, we require information concerning the physical handling of the device, such as the time required to unload and load a cartridge. While this cannot be measured precisely, due to human intervention, a reasonable estimate can be given which assumes an experienced operator. With the

COMPUTER SPECIFICATIONS

PART 6 - CARD EQUIPMENT

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
CS 601	State effective card reader speed, channel load time*, and processor usage including confirmation of image under demand conditions. CARD READER SPEED (Rightmost column read) 10 20 30 40 50 60 70 80	Cards per minute
CS 603	For card reader ____________: State time within which the next instruction must be given (executive) to avoid missing clutch points.	msec.
CS 604	For card reader____________: State time between clutch points.	msec.
CS 605	State effective card punch speed, channel load time* and processor usage including confirmation of image under demand conditions. CARD PUNCH SPEED (Rightmost column read) 10 20 30 40 50 60 70 80 CHANNEL LOAD TIME (Rightmost column read) 10 20 30 40	Cards per minute

Figure 12. Sample of Card Equipment Specifications

extensive use of random access devices in modern computing installations, these specifications can be crucial to the performance estimates of the system, and should be completed carefully and accurately. Examples of the random access specifications are given in Figure 13.

4.3.2 Extended Machine

The Computer Specifications discussed above provide for an accurate, unbiased portrayal of a complete computer system. All aspects of the physical system are covered in depth in these specifications. Modern use of computers, however, almost always presupposes that there is a software system or systems to complement the hardware configuration. Software is, in fact, a resource which affects at least two resources in an installation, the computer and the personnel involved in programming and operating the system.

Consider at this point, then, the software as it interacts with the machine to effectively modify the performance of the machine. This hardware/software interaction has been termed the extended machine.[10] The terminology is itself a useful concept, as it arose from the notion that a software system is nearly always designed to extend the use of a computer by facilitating human communication with the machine. That is, it makes the computer system more accessible to the user. The name "extended machine," then, is designed to convey the multiple effects of software systems.

The concept of the extended machine is critical in measuring the effect of a software system on the machine itself, as noted above. This measure is derived from the operating time required to perform a set of macroloops which are corollaries of some of the Computer Specifications. Appendix II of this report is the complete set of specifications which have been designed to date for the extended machine. In Figure 14, the notion of the extended machine is illustrated by a Venn diagram. Note that the extended machine is defined as the union of the computer and software systems, symbolically written

$$\text{Extended Machine} = C \cup S$$

The specifications which define the operating time of the extended machine are represented by the intersection of the computer and software systems. Thus,

$$\text{Extended Machine Operating Time} = C \cap S.$$

---

[10] The specifications for evaluating the effect of a software system on the other installation resources will be presented later in this section.

COMPUTER SPECIFICATIONS

PART 7 - RANDOM ACCESS DEVICES

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 701	Effective access times[1] and channel load times[2] in microseconds for random access device. Fill out the tables below for the following conditions: Random records[3], known machine address[4], throughput maximized, and minimum programming[5].	

EFFECTIVE ACCESS TIMES

File Size Char. / Record Size	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

CHANNEL LOAD TIMES

File Size Char. / Record Size	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 709	Maximum number of characters which can be transferred on a single request (if less than 2,000)	
CS 711	Is the file physically removable from the file drive (e.g., portable discs, cartridges, etc.)?	yes/no
CS 712	If CS 711 is yes, state the time involved in changing the file unit, (i.e., time required to unload a unit and load a new one).	

Figure 13. Sample of Random Access Device Specifications

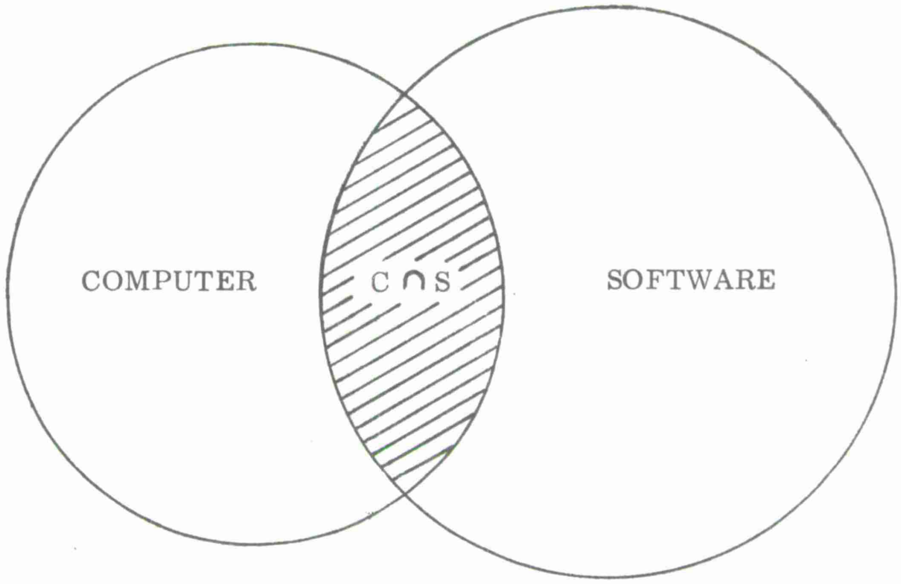

Figure 14. The Extended Machine

It is intended that these specifications be completed by the software supplier - the computer manufacturer, an independent organization, or the user's installation. Thus, each specification should be coded in the source language of the software system, compiled or assembled, and then executed on the target machine. The execution time is the response to be entered on the specification form. An implied but crucial item to note is that for each set of extended machine specifications a single target computer and a single software system are assumed. If more than one software system is used within an installation (or, more precisely, on a single computer), the extended machine specification must be completed for each software system under consideration. If, in the case of a software system in the design phase, execution time cannot be determined precisely, estimates should be made and so stated in the response column of the form.

Actually, two performance measures are derived from the extended machine specifications. First, the effect a software system has on the computer performance itself is measured. The aggregate ratio of the response for the hardware corollary of each extended machine specification over the response for the extended machine is formed. This measures the efficiency of the execution time for each macrofunction in the extended machine as compared to the execution time required to execute the hand-coded hardware corollary. Second, the extended machine specifications explicitly define some of the elements used to compute running time for an application, when a software system is used.

Appendix II of this report includes the detailed specifications for the extended machine. Note that particular attention has been paid to developing specifications dealing with central processor times. Note the column headed "Type". This column designates whether the software system for which the specification was designed is a compiler, assembler, or operating system.

4.3.3 Problem Specifications

The extended machine provides a complete, comprehensive view of the computer system and its associated software. In order to derive an estimated running time for the system in any meaningful sense, it is necessary to have a problem or application to be executed. In the classification discussion earlier in this section it was noted that certain characteristics of the task and installation lead to the selection of one or more of the five application families: dynamic file processing, static file processing, numeric computation, non-numeric processing, and/or on-line process control. In the VECTOR process mentioned earlier, the specifications of static file processing problems were provided. AUERBACH Corporation chose to extend the problem library to include dynamic file processing specifications. In effect, this now gives the Air Force two sets of problem specifications which can be used directly.

Primarily, the dynamic processing application family was chosen for additional study because it represents a wide range of problems faced by the Air Force, such as Command and Control, Inventory Control and Logistics, Information Storage and Retrieval, etc. Thus, it can be widely applied throughout various Air Force Commands. A secondary reason is that the inclusion of random access device specifications in the computer specifications naturally leads to considering the utilization of these devices.

It is important to note the usefulness of the program library concept mentioned above. In all of the problem specifications developed by AUERBACH Corporation, the analyst actually develops the specifications from the questionnaire. The problem specifications are applicable to many applications, but they require specific responses. For example, we are concerned with the number of transaction files to be processed in an application. The representative problem does not say that the timing estimate should be based on updating a master file against a fixed, predetermined number of transaction files. Rather, the number of transaction files is a variable to be stated by the analyst. The necessary guides are provided so that the analyst must include all necessary information to accurately portray the specific application.

Appendix III of this report contains the specifications for the application families dynamic file processing and static file processing. These specifications are straightforward, and should be easily completed by anyone familiar with the application at hand. As in the case of the Computer Specifications, the Problem Specifications are divided into several parts. Part 1 is very general, and really determines whether the application calls for magnetic tape or random access processing.

Parts 2 and 3 deal with the specifications for transaction files. Part 2 is general in nature, and is designed to determine such information as the storage media, number of files, etc. Part 3 is more detailed, and includes information about the number of records in a transaction file and the number of updating operations required, for example. The specifications are well defined in each case, and the analyst should have no trouble in completing them. Examples of Parts 1, 2 and 3 of the Problem Specifications are shown in Figure 15.

Parts 4 and 5, illustrated in Figure 16, concern the master file. Part 4 is general information and Part 5 requires details of the master file. The forms and the specifications for Part 5 look similar to those used in Part 3, but they should not be confused.

In Parts 6 and 7, the report file is specified. Part 6 is general, dealing with the number of report records, the format of the records, types of reports, etc. In Part 7, more details are required, such as the number of characters per report, type of report media, etc. The specification which details the report media is important, because it allows the analyst to state whether the report is to be displayed via the CRT, etc. This specification is designed particularly to provide expansion of the analytic capabilities of the AUERBACH installation effectiveness evaluation procedure to include varied types of computer (especially peripheral) equipment and varied types of applications which fall within the application families noted in Paragraph 4.2 of this report.

PROBLEM SPECIFICATIONS

PART 1 - GENERAL

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
PS 101	Is the application suited to magnetic tape oriented processing?	yes/no
PS 102	Is the application suited to random access processing?	yes/no
PS 103	If this is a random access application, state objective as either (a) maximize number of records processed or (b) minimize access time to a specific record.	
PS 201	No. of transaction records per cycle (standard).	
PS 202	No. of transaction records per cycle (peak).	
PS 203	Will the transactions be sorted in main file order?	yes/no
PS 204	Will the transactions already be on magnetic tape?	yes/no
PS 205	Will the transactions already be on the random access device?	yes/no
PS 320	No. of characters (including alphabetic, numeric, and special characters) per record.	
PS 330	No. of numeric digits per record.*	
PS 340	Average number of active fields per records (an active field is one which is used or referred to duringp rocessing).**	
PS 361	State the number of files accessed per transaction: read, write, or write and check usages of the random access records if necessary during the processing.	
PS 362	State the number of read, read/write, and read/write/check usage of the random access records during the processing.	

Figure 15. Sample General Problem and Transaction File Specifications

PROBLEM SPECIFICATIONS

PART 4 - MASTER FILE

(Use 1 per Master File)

SPECI- FICATION NO.	SPECIFICATION	RESPONSE
PS 401	No. of records in the master file.	
PS 402	No. of major record types in the file.	
PS 403	What are the time intervals at which random records must be updated to remain usable?	
PS 510	No. of records of this type in the Master File.	
PS 520	No. of characters (including alphabetic, numeric, and special characters) per record.	
PS 530	No. of numeric digits per record. *	
PS 540	Average number of active fields per record (an active) field is one which is used or referred to during processing). **	
PS 560	No. of Simple Field Updates or Equivalent Operations per record. (This is the equivalent of the sum of the add/subtract and comparison operations needed to process a record.)	
PS 570	No. of Complex Field Update Steps or Equivalent Operations per record. (This is the equivalent of the sum of multiply and divide operations per record.)	
PS 580	Average no. of decimal digits in the numeric operands used during the process.	

Figure 16. Sample Master File Specifications

Part 8 of the Problem Specifications details the specifications for each printed report. For example, information is required concerning the number of lines in headings, body, and footings of each report, as well as width and length of the printed form, etc. Figure 17 illustrates Parts 6, 7, and 8 of the Problem Specifications.

It is important to note that separate specifications (Parts 6, 7, and 8) must be completed for each different report. This will insure an accurate appraisal of the estimated running time for the representative problem, since peripheral equipment such as printers are frequently limiting factors.

The estimated running time for the representative problem is derived by algorithmically combining the Computer and/or Extended Machine Specification and Problem Specifications. This is the essence of the VECTOR process and will not be discussed in detail in this report. It is important to note, however, that the Extended Machine Specifications will be substituted for their hardware corollaries in the VECTOR algorithms, if these specifications are available.

4.3.4 Software Specifications

One aspect of software evaluation has been considered through the concept of the extended machine. As stated previously, software also is an important resource to consider in its effect on installation personnel. In Appendix IV of this report, we have included a set of software specifications designed for this purpose. The software specifications are divided into six parts, which are relatively independent of each other. They are:

Part 1 — Diagnostics
Part 2 — Program Structuring Elements
Part 3 — Storage Allocation and Protection
Part 4 — Program Library Facilities
Part 5 — Maintenance, Modification, and Documentation
Part 6 — Training and Familiarization

It will be noted that scoring rules are included with each specification. This precludes another pass through the specifications to arrive at an effectiveness

PROBLEM SPECIFICATIONS

PART 6 - REPORT FILE

(Use 1 per Report File)

SPECIFICATION NO.		RESPONSE
PS 601	No. of report records per cycle (standard).	
PS 602	No. of report records per cycle (peak).	
PS 603	Should the reports be sorted in main file order?	yes/no
PS 604	May the reports be placed on magnetic tape for off-line printing?	yes/no
PS 710	No. of records of this type in the Report File.	
PS 720	No. of characters (including alphabetic, numeric, and special characters) per record.	
PS 730	Report Media (Communication line, CRT display, Hard Copy, etc.)	
PS 810	Width of printed form in number of characters.	
PS 820	No. of printed alphanumeric lines per physical form.	
PS 830	No. of printed numeric lines per physical form.	
PS 840	No. of printed lines per heading.	
PS 850	No. of printed lines in report body.	
PS 860	No. of printed lines per footing.	

Figure 17. Sample Report File and Printed Report Specifications

measure. Since the scoring rules are included in the specifications, it is felt that impartial Air Force personnel or an independent consultant should be assigned to complete these specifications.

Since the various parts of these specifications can be considered separately the overall view is that the highest possible score for all specifications is 44. Determination of tolerable limits for each part will be left to Air Force management, since these can vary.

In Part 1, several types of diagnostics are included, all of which are important to the user of the system. For program checkout (i.e., debugging), three basic types of diagnostics are considered: snapshot, trace, and postmortem diagnostics. These different types can exist in various combinations in any given system, and the software analyst should state the particular permutation which exists and score accordingly. One should also be concerned with diagnostic facilities for the source language. In particular, the level of syntactic diagnostics available to flag errors in the source (input) language at compilation or assembly time is important. For example, a compiler should have the ability to scan a statement like

```
X = ( (A + B) / (C + D)
```

and flag the error as

```
EXTRA LEFT PARENTHESIS.
```

In most software systems it is also useful to have diagnostic facilities which can flag compilation or assembly errors which result in attempts to overrun available memory, for example, with a message like "COMPILED PROGRAM EXCEEDS AVAILABLE MEMORY." Part 1 of the Software Specifications includes specifications in all three of these areas of concern.

Part 2 is a set of specifications for program structuring elements. For example, specifications have been prepared which deal with levels of subroutine nesting, types of subroutines permitted, etc. In addition, there are specifications which deal with the inclusion of linguistic macro-type entries to perform some of the common programming techniques such as iteration, conditional tests, etc. An example of iteration expressed in FORTRAN IV is

```
    DO 4 I = 1, 10, 1
4   A = A + I
```

where the underlined 'DO' is the linguistic device which expresses the iteration.

In Part 3, specifications are detailed which cover facilities for dynamic storage allocation and protection. Included in these specifications are systems programs such as the compilers, operating system, etc., as well as the users' programs. Data storage and protection refers to the users' data, as well as general systems data; i.e., the users' program is considered to be data to the compiler or operating system. Thus, attention is directed to an automatic storage mapping function, to relieve the programmer of this particular chore and also program relocation activities.

Program library facilities are very important in most installations. Part 4 of the Software Specifications defines a set of program libraries as being complete, intermediate level, and minimum. While the main concern is with the existence of these facilities at the time the software is first considered at an installation, provision has been made for scoring those systems in which the library facilities are developed after the fact, so to speak.

Maintenance, system modification, and documentation have in the past been difficult areas for the user. Completion of the specifications in Part 5 of the Software Specifications will often reveal to the user potentially troublesome areas. SS507 is probably the least straightforward of these specifications and deserves some comment here. This specification deals with the retention of the users' source language program by the system, either on magnetic tape or in a disc file. This facility can provide the user access to his source language for checkout or updating purposes without going through a recompilation or reassembly process, and also avoids the problem of saving many versions of the program in hard copy. This is a feature which is not commonly provided today, but it is a useful notion which will probably receive more emphasis in the future.

Part 6 of the Software Specifications is designed to cover two aspects of training in the use of the software system under consideration. First, formal training sessions by the supplier are considered. There should be two aspects to this training, the source language and the practical use of the system. The score is determined by the amount of training provided. If training in both aspects is offered, the score will be higher. This amount of training is very important to the installation which will use the system, since it enables programmers to become proficient in its use more quickly.

Even after formal training has been completed, a certain amount of time is required before programming personnel become at ease with the system. Thus in specifications SS704-706, one should consider how long it takes a programmer to learn the tricks and idiosyncrasies of the system. Consideration should also be given to the time required to learn through experience which constructs tend to slow down the object program, and to avoiding them where desirable.

The term "appropriate personnel" used in these specifications refers to junior or journeyman programmers, senior programmer/analysts, and lead programmers. Not all personnel may be required to use a particular software system. For example, one would not expect a junior programmer to use a sophisticated system designed for use by senior people or well-versed specialists. This should be taken into consideration when these specifications in Part 6 are completed.

It should be noted that AUERBACH Corporation does not claim that these software specifications are complete, or that they answer every specific need. However, those specifications have been developed for which answers are usually available to the user. Many other specifications can be listed, but it is not known at this stage of development what the implications of the responses might be. This type of specification has been avoided until such time as it is known how to use the response in effectiveness evaluation. It should also be noted that the user can add specifications which are of particular value in a given situation. Similarly, these specifications or even major parts as defined above may be deleted, if they are not germane to the application or installation. The effect of such changes on the scoring system should be taken into account when they are made.

4.3.5 Installation Operating Specifications

The operating specifications or the items to be measured in the operation of a computer installation have been divided into some seventy or more data elements to be collected as part of a daily operation of a computer installation. This is an attempt to establish procedures for the collection of management data. From the initial collection of this management reporting data and subsequent iterations in data collection achieved through refinement of the operating specifications, AUERBACH hopes to be able to provide management, at both the operating and policy making levels, with sufficient information to determine those elements of computer installation resource allocation that are critical. The identification of these critical elements will result from collection of the data outlined in the installation operating specifications, combining of these quantitative values under a data conversion algorithm, operating on the data via the statistical technique known as stepwise multiple regression analysis, and using this statistical technique to determine the relative significance of each of the factors presented to the total operating efficiency of the installation.

Part 1 refers to the collection of data relative to numbers and salaries of personnel. Part 2 is devoted to the cost and utilization factors incurred in operating installations through expenditure of such resources as maintenance, supplies, and square footage of space. Part 3 is devoted to the equipment cost and equipment utilization in terms of hours and number of runs per month. For the purposes of this questionnaire, a run is defined as a unit of work in which a discrete starting point and discrete ending point can be defined. This definition is utilized to account for isolated random processing of tasks and to incorporate each of these either as a single run (if only one item was processed for the day) or as an element of a run (if many related items were processed for the day, so that all related runs for a dynamic file processing application over a 24-hour period could be counted as one run). This element is validated by the fact that data is collected on the number of hours for production runs per month and over the course of a month's time, these factors should provide a reasonable average.

Parts 4 and 5, devoted to installation and program performance respectively, are aimed at collection of data on a project basis. It is our opinion that this is the most meaningful orientation for collection of data, since in a given installation any number of different families or application types may be utilized. Our classification scheme referred to earlier in this section is geared to this division of tasks at the project level.

It is believed that the information presented in Parts 1 through 5 of the installation operating specifications covers the major itmes of significance relative to the utilization of resources within the computer installation. Resources are defined as such quantitatively measured items as computer equipment, number and cost of programming and operating personnel, and supplies. In addition to these measurements of the resource allocation, one must define the criteria for efficient utilization. These criteria are stated in Parts 4 and 5 of the installation operating specifications. We recognize that, particularly in Sections 4 and 5, many of the data items that have been requested for quantitative completion will not be available in any quantitative form within the installations to be examined. It is intended, where this data is not available, to examine the records maintained in whatever form that should happen to exist to compile this data. Where such examination does not yield any pertinent result, averages of combined best-estimate information will have to suffice for the initial gathering of data.

It is further believed that, by imposing these procedures for collection of operating statistics within Air Force installations, a by-product service is being created which will be of great value to Air Force management. The standardized collection of data as defined in these installation operating specifications will force all installations to collect data in a uniform manner. This of itself will provide a basis for valid comparisons of operating procedures.

Furthermore, the use of these procedures for collecting data will force planning of new projects to include provisions for collection of data according to the definitions that have been set up in the questionnaire. Again, this has the beneficial effect of standardizing management of resources such as programmers and project management within a group of installations. Eventually, sufficient data should be available to allow classification of data into the various computer installation environments and problem types to provide standards that can be used as guides for allocation of resources to project tasks.

4.4 RESOURCE ALLOCATION

The specifications and concepts discussed in Paragraphs 4.2 through 4.3.5 all provide some quantitative measurement of the resources available to a computer installation. The resources which have been identified are:

(1) The computer system(s) and its extended machine performance produced by the interaction of the software system on the raw hardware performance. The extended machine performance is used primarily to produce an estimated computer performance time. The extended machine specification is processed through a conversion algorithm to produce a series of data elements that significantly affect the computer running time. In turn the problem specification, a representative set of problem specifications for the application family, is merged with computer data elements to produce an estimated computer running time for a given set of problem parameters (e.g., volume, file size, transaction activity, etc.).

(2) The software system(s) contributes to the effective utilization of programming and operating personnel by its effect on the amount of training time spent by programmers in learning a new computer system, the aid it provides in shortening problem analysis, the amount of by-product documentation produced, as well as the built-in check-out facilities, all of which have a pronounced effect on the amount of time spent by programming personnel on each of these activities.

In turn, computer operating systems will play an increasing role in determination of the duties of operator personnel and the number of personnel necessary for each shift.

(3) Installation Operating Specifications have been divided into five sections:

(a) Personnel represent perhaps the single largest factor in determining overall systems performance in terms of dollars. Personnel have been categorized as:

1. Programmers
2. Operators
3. Clerical
4. Managerial
5. Administrative (support rather than direct)

It is intended to measure both numbers and costs of each of these categories relative to the volumes processed, installation environments, and application families that exist within the installation.

(b) Data is to be collected on equipment utilization to compare utilization to the de facto standards derived for each representative problem type for each application family. Counts of the total number of program runs and checkout runs per month serve as validity checks on data collected in Part 3 (utilization of supplies), Part 4 (delivery of projects within time and budget estimates), and Part 5 (number of hours spent on checkout). Additionally, the data will be used to provide guidelines on how much machine time is necessary for checkout of particular application families under varied installation environments. Eventually, for example, it should be possible to predetermine the number of hours of checkout necessary for a 50,000-step inventory control problem using batched input in a centralized operation.

(c) Data collected on maintance and supplies utilization is used to derive guidelines as to the normal usage of these resources under varying conditions of application demands, installation environments, and management policies.

(d) Data collected on project performance is used to indicate the relationship of the effective utilization of men, equipment, and supplies to delivery of results. It provides management, at the operating level, with a measure of how much effect the utilization of more of any resource item (i.e., more or higher level programmers, more machine check-out time, more elaborate software) can affect the delivery and cost of each project.

When such data comparisons are extended over a range of installations working on similar application tasks in similar environments, they provide a standard.

(e) Collection of data on programmer time is most directly related to the single, most controllable variable in the performance evaluation, namely programming costs. These costs are collected separately on each project and within each application family and classified by a large number of factors including such items as state of problem definition, number of changes in problem requirements, simplicity of problem logic, interface with other programs, etc. Since it would be difficult to place a quantitative value on each of these contributing factors, it was determined that the time expended on each categorized activity would, over a series of projects, provide the most accurate measure of the contribution of that activity to overall programming cost. It may be revealed that non-numeric processing problems require more logical analysis time than inventory control or personnel accounting, as a result of an analysis of the time spent on this category over a range of problems. Resource allocation analysis will reveal these relationships.

4.4.1 Software as a Resource

AUERBACH has identified software as a separate resource, since it affects both computer hardware operation and programmer performance. Details of the effect of software on computer hardware have been described in Paragraph 4.3.2. The software specifications for determining its impact on programming personnel and management have been discussed in Paragraph 4.3.4. The rating of the software system in both areas provides a good indication to management of the impact of the software system on the installation as a whole. Furthermore, these measures are used in the multiple regression analysis. This will ensure that this important resource is duly considered in the installation effectiveness evaluation, and that software, which is itself a dynamic force within the ADP community, will be a part of the dynamic, self-improving effectiveness evaluation technique developed in this study. Both of these aspects of software evaluation will help to develop more vigorous standards for the use and measurement of software systems and the total installation.

4.4.2 Installation Performance Summary

All of the resources referred to in Paragraph 4.4 are to be considered part of the AUERBACH effectiveness evaluation procedure. The first step after an evaluation is completed is to inform local management of the results of the analysis. This is accomplished by reporting to that level of management its position with respect to machine utilization, based in part on estimated versus actual running times for representative tasks, personnel utilization measured in part on the same basis, estimated (target) delivery time versus actual delivery time, etc. Many other items of specific interest to installation management will also be presented in the installation summary. A sample of this summary report is indicated in Figure 3.

The information contained in the installation summary report will be further analyzed through stepwise multiple regression analysis, and results will be compared with similar installations according to the standards developed in the AUERBACH procedure. These further results will be presented to higher level Air Force Management in the form of an installation profile, which will indicate the strengths and weaknesses of a particular installation or proposed installation. This profile can be used by higher level management in its resource allocation and planning as dictated by Air Force and Department of Defense needs.

The sample Installation Summary Report shown in Figure 18 includes information of direct value to the installation management. In particular, allocation of the installation's major direct resources, equipment, software, and personnel is summarized with respect to scores derived in the processes described above in Paragraphs 4.1 through 4.3.5 and combined in the resource allocation measurement algorithm.

The Installation Summary Report is divided into six major sections:

(1) General

(2) Equipment

(3) Software

(4) Personnel

(5) Cost

(6) Accomplishment

These sections are closely related to the format suggested by the Bureau of the Budget[11] in the Chapter entitled "Information for Managing Automatic Data Processing Activities." The sections are not necessarily mutually exclusive, since all of the resources of a computer installation are related. The utilization of each resource is determined as independently as possible.

4.5 STANDARDS

The collection of computer and problem data, task and installation classification data, and the manipulation of installation performance data is not the end of the evaluation process. It is, in fact, only one step in the process. As in any evaluation, the evaluator or reviewer must compare what he knows about that which is being evaluated to some kind of standard. The standard may be some officially promulgated quantitative value, it may be a qualitative but generally accepted policy, or it may be an unstated but intuitively sensed "standard" in the mind of the reviewer. In any case, it is the comparison of performance data against performance standards which permits performance evaluation.

Task and installation classification provides a basis for standards development. It is meaningless to compare installations which fall into different categories

---

[11] Ibid., pp 65-66

unless they have a common denominator such as is provided by the functional application families. Similarly, it is rather shortsighted to compare solely on the basis of the application family, since the applications could be performed under divergent circumstances, e.g., they could utilize machines of varying capacity in terms of speed and memory size.

As noted above in Paragraphs 4.1 through 4.3.5, the AUERBACH effectiveness evaluation procedure collects data concerning all facets of an ADP installation. The algorithmic combination of data in the form of computer and problem specifications leads to an estimated running time for a particular application family. This estimated running time represents a de facto standard for comparison.

The AUERBACH performance effectiveness evaluation procedure provides for a set of standards which are consistent yet dynamic, are subject to statistical validation, and may be modified easily to meet the needs of the user. In general, the standards are empirically developed, but the development process never ends.

All of the data collected as installation operating characteristics and resource allocation statistics contributes to the development of standards and the subsequent updating process. Although the precise method of handling the data cannot be determined until actual performance and classification data is collected from several installations, the approach is quite clearly identified.

The standards are based upon the data collected in the application of the AUERBACH process. For each value, raw or manipulated into a ratio or formula, there may be a standard. The standard, could be in the form of an average of all collected values or the median, or, depending upon the distribution of values, it could be indicated as a range of statistical significance. The raw data specifications will be refined or summarized through standard procedures to be developed and documented (see Section III) in order to provide input to a procedure for determining the relative significance and index value to be associated with each performance measurement criterion. The standards will be measured by statistical techniques and in each case the method offering the most significant standard will be adopted.[12]

---

[12] For a discussion of the statistical techniques and the method of their application, see Paragraph 4.6.

As has been implied many times in this report, it can be expected that some measurements could have exceptionally wide ranges, since there are many important differences among computer installations. It is for this reason that an installation and task classification scheme has been developed and that, in practice, installations should be compared to installations of the same class. Consequently, it is anticipated that most of the standards will be similarly classified so that the results of an installation evaluation will be as meaningful as possible.

The dynamic nature of the standards aspect of the AUERBACH process comes about through continued use of the process. As installations are evaluated and as data is collected, more and more data becomes available for refining and, perhaps, making major changes in standards. The statistical analyses referred to above will be repeated from time to time and it is anticipated that the standards will become more realistic and consequently more useful as time goes by.

Furthermore, anticipating continued rapid changes in the state-of-the-art of computing, it is essential that the evaluation process be capable of being modified as the nature of installations, applications, and computers changes. Since there should be a constant input of new data into the system, it is conceivable that the standards refining analyses will be able to detect trends and will highlight areas in which the impact of aged data should be reduced. This dynamic, self-correcting feature of the AUERBACH process not only reduces the likelihood of early obsolescence of the process, but should actually cause it to improve with age.

4.6 STATISTICAL ANALYSIS

Validation of the AUERBACH process for effectiveness evaluation will be accomplished statistically. A mathematical model will be constructed and used to:

(1) Verify the appropriateness of the classification scheme developed to categorize installations, performance data, and standards.

(2) Determine the relative significance of the data elements used in the evaluation process.

(3) Provide a rigorous method for combining the fundamentally dissimilar information collected and reducing it to a form directly usable in quantifying effectiveness.

(4) Introduce a self-refining quality into the AUERBACH process to assure the modification of the process as external factors induce a need for modification.

The principal statistical tool to be used is stepwise multiple regression analysis.

4.6.1 Stepwise Multiple Regression Analysis

Multiple regression analysis is a technique used in data analysis to obtain the best fit of a set of observations of independent and dependent variables by an equation of the form:

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n$$

where y is the dependent variable; x_1, x_2, ... are the independent variables; and b_0, ..., b_n are the coefficients to be determined.

By means of a least squares fit for a particular set of observations, a set of coefficients is derived for the dependent variables. The solution also provides:

(1) A measure of reliability of each coefficient.

(2) The degree to which the fit approximates the assumed relationship.

(3) The contribution of each dependent variable to the relationship between the set of dependent variables and the independent variable.

(4) The degree to which the set of dependent variables explain the variation in the independent variable.

(5) The degree of reliability of the relationship between dependent and independent variables.

(6) The statistical significance of this relationship.

A modification of the regression approach is known as the stepwise method. In the stepwise procedure, intermediate results which are not even recorded by normal calculation methods are used to give valuable statistical information at each step in the calculation. These intermediate answers are also used to control the method of calculation. Essentially, without adding greatly to the number of arithmetic steps, a number of intermediate regression equations are obtained, as well as the complete multiple regression equation. These equations are obtained by adding one variable at a time and thus give the following intermediate equations:

$$y = b_0 + b_1x_1$$

$$y = b_0' + b_1'x_1 + b_2'x_2$$

$$y = b_0'' + b_1''x_1 + b_2''x_2 + b_3''x_3$$

.

.

.

The variable added is that one which makes the greatest improvement in "goodness of fit." The coefficients represent the best values when the equation is fitted by the specific variables included in the equation.

An important property of the stepwise procedure is based on the facts that (1) a variable may be indicated to be significant in any early stage and thus enter the equation, and (2) after several other variables are added to the regression equation, the initial variable may be indicated to be insignificant. The insignificant variable will be removed from the regression equation before adding an additional variable. Therefore, only significant variables are included in the final regression.

In summary, therefore, the stepwise multiple regression technique enables an investigator to hypothesize the impact of a large set of dependent variables on the independent variable and test:

(1) Which ones are indeed significant.

(2) The degree of variation explained by these significant variables.

Based on the relationship for each group studied, the analysis will predict the value of the independent variable to be used as a standard.

4.6.2 Example of Stepwise Multiple Regression Analysis

A set of questionnaires has been designed as indicated in the previous sections. These questionnaires contain data from each installation, such as:

(1) Number of Analysts

(2) Number of Programmers

(3) Amount of Square Footage Utilized by Installation

(4) Number of Machine Operators

(5) Complexity of Task

(6) Number of Program Steps

(7) Number of Checkout Runs

This data will be correlated by means of Stepwise multiple regression with the number of dollars expended by that installation. The stepwise regression analysis by an iterative process will select the relevant variables and will weight them in such a way that the relative importance of each variable will be indicated. For example, assume that the iterative process has selected variables (6) and (7) as noted above (number of program steps and number of checkout runs respectively). These variables are then weighted by the procedure indicated in the following calculations. The following table indicates the basic data collected by means of the questionnaire:

Installation (N)	No. of Program Steps	No. of Check-out Runs	$ Expended
1	20	20	500
2	30	10	400
3	40	40	800
4	20	10	300
5	40	20	600

Step 1. The regression equation that would be used to fit the variables to an equation is:

$$y = ax_1 + bx_2 + c$$

The desired values of a, b, and c are such that the sum of squares of the errors between actual and normal y-values is a minimum. The normal equations are obtained by differentiating with respect to a, b, and c respectively, the expression:

$$\sum (y - ax_1 - bx_2 - c)^2$$

After setting each of the first derivatives equal to zero and simplifying we arrive at the normal equations:

$$\sum x_1 y = a\sum x_1^2 + b\sum x_1 x_2 + c\sum x_1$$
$$\sum x_2 y = a\sum x_1 x_2 + b\sum x_2^2 + c\sum x_2$$
$$\sum y = a\sum x_1 + b\sum x_2 + Nc$$

Step 2. The values necessary for the solution of the normal equations are then computed from the following table.

Installation (N)	$ y	No. of Program Steps	No. of Check-out Runs	x_1y	x_2y	x_1x_2	x_1^2	x_2^2
1	500	20	20	10,000	10,000	400	400	400
2	400	10	30	4,000	12,000	300	100	900
3	800	40	40	32,000	32,000	1,600	1,600	1,600
4	300	10	20	3,000	6,000	200	100	400
5	600	20	40	12,000	24,000	800	400	1,600
Total	2,600	100	150	61,000	84,000	3,300	2,600	4,900

These values are as follows:

$$N = 5 \qquad \sum x_1^2 = 5,400$$
$$\sum y = 2,600 \qquad \sum x_2^2 = 2,600$$
$$\sum x_1 = 150 \qquad \sum x_1 y = 89,000$$
$$\sum x_2 = 100 \qquad \sum x_2 y = 61,000$$
$$\sum x_1 x_2 = 3,500$$

Step 3. Substitute these values into the normal equations and solve simultaneously following the procedure outlined below:

(1) $61,000 = 2,600\ a + 3,300\ b + 100\ c$
(2) $84,000 = 3,300\ a + 4,900\ b + 150\ c$
(3) $2,600 = 100\ a + 150\ b + 5c$

Divide each equation by the coefficient of c.

(4) $610 = 26\ a + 33\ b + c$
(5) $560 = 22\ a + 32.66\ b + c$
(6) $520 = 20\ a + 30\ b + c$

Subtract (5) and (6) from (4) successively.

(7) $50 = 4\ a + 33\ b$
(8) $90 = 6\ a + 3\ b$

Divide by the coefficient of b.

(9) $150 = 12a + b$
(10) $30 = 2a + b$

Subtract equation (10) from equation (9).

$$(11) \qquad 120 = 10a$$
$$a = 120/10 = 12$$

Substitute in (8) and solve for b.

$$90 = 6(12) + 3b$$
$$b = 6$$

Substitute the values for a and b into equation (6) and solve for c. (Note that a, b, or c may be negative, but this does not affect the usefulness of the method.)

$$520 = 20\ (12) + 30\ (6) + c$$
$$100 = c.$$

Substitute the values for a, b, and c into equation (3), getting an identity (2600 = 2600) proving the value obtained. The equation is of the following form:

$$y = 12\ (\text{No. of Program Steps}) + 6\ (\text{No. of Checkout Runs}) + 100$$

This equation is then used in the determination of standards for other and subsequent installations. This requires that a system be employed to record the output variables and the dollars expended.

Step 4. Assume that for the following the data below is collected:

Installation (N)	x_1	x_2	Actual $ Expended
1	20	10	500
2	5	25	500
3	30	5	500
4	15	30	500
5	10	20	500
	80	90	2500

These output figures are put into the formula to determine standard expenditures.

Installation (N)	Standard $
1	400
2	310
3	490
4	460
5	340
	2000

The performance rating of the installation is then determined by dividing standard dollars for each installation by actual dollars, i.e., 400/500 or 80 percent for installation 1 as shown in the last two tables.

The 80 percent performance measure calculated by multiple regression analysis integrates the innumerable variables of this particular installation, affording management a useful tool for action.

This has been accomplished by:

(1) Analyzing the variables which account for most of the variability in output.

(2) Correlating outputs to these variables by means of stepwise multiple regression analysis.

(3) Comparing actual dollars to standard dollars as determined by the derived formula.

4.6.3 Application of Statistical Analysis to Effectiveness Evaluation

The primary model will relate the total cost of operating the existing or proposed installation to a set of readily identifiable variables. This set of variables will be originally selected from those appearing on the various data collection documents contained in the Appendices to this report and illustrated above. The original set will be developed intuitively; it is desirable - but far from critical - that the most significant variables be selected at this point.

Data will be collected from a sample of presumably homogeneous installations performing nearly homogeneous tasks (e.g., dynamic file processing). Each of the selected variables will be scaled. The variables will then be analyzed by means of the stepwise multiple regression technique.

The output of such an analysis would be:

(1) A weighting factor (coefficient) for each variable.

(2) The degree of relationship between the weighted variables and dollars expended. If the degree of relationship is statistically significant, then the significant variables and their relative weights do indeed explain the dollar variation. If this, however, is not the case, other variables can be tested. This provides a mechanism for validating the variables hypothesized.

(3) A preliminary standard for each installation based on the variables uncovered as significant and their relative importance.

(4) A comparison of the actual dollars expended to the calculated standard dollars, revealing the degree of deviation of the actual from the standard.

(5) The relationship of the significant variables and the dollars expended expressed in terms of an equation. The sensitivity of each variable can be tested by means of a sensitivity analysis to study the impact of the variables on the dollars expended. As a result, the relative impact of the variables commensurate with their values can be ascertained.

This process will be repeated many times until the correlation indices indicate that the model does in fact describe the real world, within statistically acceptable limits. The number of iterations that will be required cannot be predicted at this point; however, convergence can be expected - i.e., the outputs of each iteration will normally be better than all prior outputs.

It is not required that all variables be linear. Logarithmic, exponential, or trigonometric relationships are permitted and will be tested for significance. Similarly, multivariable terms will be tested if indicated. The scales and values for the most significant variables will be revised where appropriate to increase their accuracy. Statistically insignificant variables will be eliminated as their insignificance is revealed.

Each time an installation is evaluated, the variables which are used in the model will be "plugged in" and the standard annual cost determined. The ratio of the actual (or planned or budgeted) cost to the standard will provide a comprehensive, overall performance effectiveness rating. Such a rating, along with the data reported on the Installation Performance Summary, provides the principal inputs to the final document, the Effectiveness Evaluation Report.

Furthermore, the model will be periodically revised, on the basis of additional data collected as additional installations are evaluated. As described in Paragraph 4.5 this process will refine the model and assure its continuing significance. The model will also indicate the validity of the classification scheme in use at any time; consistently low correlation implies a nonhomogeneous class, while high correlation within subsets of a class of installations may indicate a need for definition of new classes.

Consequently, as the state of the art advances, the statistical technique will be constantly used to revise and revalidate the classes and standards to reflect the anticipated increased efficiency in information processing. This will keep the evaluation procedure up-to-date and, hopefully, reveal new insights into the functioning of the world of information science.

APPENDIX I

COMPUTER SPECIFICATIONS

DIRECTIONS FOR COMPLETING COMPUTER SPECIFICATIONS

In completing the computer specifications, it should be remembered that the purpose is not to portray the machine in isolation, but rather as it will be used in the operating environment. Thus, the specifications are concerned with a specific configuration of equipment, not necessarily a maximum or minimum system.

It is anticipated that the computer specifications will be completed by a representative of the manufacturer who has access to all of the relevant facts and knowledge of the most efficient techniques for programming and operating the system. The estimated running time (and hence, in part, the performance time) for a computer system will be directly related to the responses entered in the computer specification form. Where specific tasks have been defined, it is intended that the definition serve as a guide for the programming. If the answer cannot be determined, the response should be stated as "UNKNOWN." It is of great importance that each appropriate specification response be entered as completely and realistically as possible.

The computer specifications have been divided into seven parts, each of which can be completed separately. Part 1 is a specification of fundamental computer systems characteristics, such as word size, number of words of main memory, number of processors, etc. These specifications are straightforward, and should be self-explanatory.

Part 2 deals with central processor timing information. It should be noted that raw add time, for example, as stated in an advertising brochure is not applicable here, except as part of the program loop. Specifications which indicate how the hardware above might perform certain executive functions, e.g., the time required to respond to I/O interrupt conditions, are also included in this part.

In Part 3, Input/Output process-related timing, the specifications are not so straightforward as in the other parts. This is due to the fact that input/output processing timings are derived from tasks which can be done in several ways. For example, the relatively simple task of initiating a card read instruction may require verification of the code, translation into a form acceptable to the computer, etc. There are many ways to accomplish these tasks, some of which are applicable to some computers and not applicable to others. The specifications in Part 3 allow for permutations of these various conditions. It is important to pay particular attention to these specifications since they may play a major role in determining the performance (running time) estimate of a computer system.

Part 4 is a straightforward set of specifications devoted to magnetic tape subsystem performance. Included here are specifications which define simultaneous operations such as read/compute, peak speed of the tape units in terms of alphanumeric characters per second, recording density, etc. If tape subsystems of different rated speeds are components in a computer configuration, a separate specifications form must be completed for each one.

Part 5 includes specifications for line printers such as effective transfer rates, and central processor overheads. These specifications include both timing information and physical capacity of the printer in terms of page width in number of characters, etc. There are two specifications for use in computing effective printer speed. One requests speed for a stated alphanumeric character set, and the other is for a stated numeric character set. Should the effective speed be the same in both cases, each specification should be completed with numeric answers. The specifications contained in this part are straightforward, and should be completed.

Part 6 is a set of specifications for punch card equipment (readers and punches). In the specifications for effective reader or punch speeds and channel usage time, space is provided for different figures if speeds vary according to the rightmost column read or punched. If the speed does not vary, only the 80th column is to be completed. Since card equipment is largely mechanical, specifications have been provided for time required to engage and disengage a clutch, if one is used in the device.

Part 7 is composed of the specifications for random access devices. These specifications are designed to include all types of presently known equipment. The devices have been characterized as nonremovable, e.g., a disc file, where the storage element cannot be physically removed; and removable, e.g., a cartridge unit, where the storage element can be physically removed by an operator. For nonremovable devices, the response for specifications time required to change storage elements, etc. should be marked N/A (Not Applicable). While these specifications are straightforward, they will require time to complete, since tables must be completed for distinct sets of conditions. The specifications regarding effective access time are stated in terms of file sizes, rather than rated physical capacity of the device. This should be noted in order to avoid any confusion and inadvertent misrepresentation of the random access subsystem.

The specifications for the central processor and the various tasks require completion <u>only once</u> by each manufacturer for each computer system. However, as stated above, for peripheral gear such as printers, card equipment, and random access devices, one set of specifications must be completed for each piece of equipment having different characteristics, if it is included in the configuration. For example, a computer system configuration which employs on-line high-speed and low-speed line printers requires that a specification form must be completed for each device. This is true also for card equipment and random access devices, where applicable.

COMPUTER SPECIFICATIONS

PART 1 - SYSTEM SPECIFICATIONS

SPECIFICATION NO.	APPLICABLE FOR					SPECIFICATION	RESPONSE
	BW	DW	DC	AW	AC		
CS 101	x	x		x		Main memory size in words.	words
CS 102		x		x		Word size in alphanumeric characters.	chars.
CS 103		x		x		Word size in decimal digits.	digits
CS 104	x					Word size in bits (excluding check bits).	bits
CS 105					x	Main memory size in characters.	chars.
CS 106			x			Main memory size in decimal digits.	digits
CS 107			x			No. of decimal digits per alphanumeric character.	
CS 108	x	x	x	x	x	No. of index registers.	
CS 109	x	x		x		Main memory volume that can be accessed without indexing, in words.	words
CS 110					x	Main memory volume that can be accessed without indexing, in characters.	chars.
CS 111			x			Main memory volume that can be accessed without indexing, in decimal digits.	digits

Abbreviations

BW - Binary Fixed Word Systems
DW - Decimal Fixed Word Systems
DC - Decimal Character Oriented Systems
AW - Alphanumeric Fixed Word Systems
AC - Alphanumeric Character Oriented Systems.

COMPUTER SPECIFICATIONS

PART 1 - SYSTEM SPECIFICATIONS (CONT'D)

SPECIFI-CATION NO.	APPLICABLE FOR					SPECIFICATION	RESPONSE
	B W	D W	D C	A W	A C		
CS 112						No. of input/output data channels	
CS 113						No. of line printers	
CS 114						No. of card readers	
CS 115						No. of card punches	
CS 116						No. of random access devices	
CS 117						No. of processors	
CS 118						No. of arithmetic units per processor	

COMPUTER SPECIFICATIONS

PART 2 - PROCESSOR TIMES

SPECIFICATION NO.	APPLICABLE FOR					SPECIFICATION	RESPONSE
	B W	D W	D C	A W	A C		
CS 201	x	x	x	x	x	Time taken to add two operands in main memory and store the sum (operands must have more than 4 decimal digits).	μsec.
CS 202		x	x	x	x	Time taken to multiply an X digit operand by a Y digit operand and store the product (X and Y must be greater than 4).	μsec.
CS 203		x	x	x	x	Time taken to divide an X digit operand by a Y digit operand and store the quotient (X and Y must be greater than 4).	μsec.
CS 204	x					Time taken to multiply two operands in main memory and store the product.	μsec.
CS 205	x					Time taken to divide two operands in main memory and store the quotient.	μsec.
CS 206	x	x	x	x	x	Time taken to index in operand.	μsec.
CS 207	x	x	x	x	x	Time taken to compare 2 operands in main memory (of at least 8 decimal digits or equivalent) and to transfer control to one of two arbitrary locations based on the result of the comparison.	μsec.

COMPUTER SPECIFICATIONS

PART 2 - PROCESSOR TIMES (CONT'D)

SPECIFICATION NO.	APPLICABLE FOR					SPECIFICATION	RESPONSE
	BW	DW	DC	AW	AC		
CS 208	x	x	x	x	x	Time taken to perform the following task: A 1-digit operand, whose value is 1, 2, 3, 4, 5, or 6, is held in main memory. This is used to transfer control to one of six locations. The time stated includes a check that the value is between 1 and 6, and all necessary work up to and including the transfer of control. If the time varies, based on the value of the data item, quote a formula.	μsec.
CS 209	x					Time taken for the following task: A four-bit operand is presently stored in the middle of a computer word. It is needed for use as a numeric operand, effectively right justified. The task is to prepare it for this use. (If no action need be taken, the time is zero.) Normally, it will be necessary to place it into another location.	μsec.
CS 210		x		x		Time taken for the following task: A single-digit operand is presently stored in the middle of a computer word. It is needed for use as a numeric operand, effectively right justified. The task is to prepare it for this use. (If no action need be taken, the time is zero.) Normally, it will be necessary to place it into another location.	μsec.

COMPUTER SPECIFICATIONS

PART 2 - PROCESSOR TIMES (CONT'D)

SPECIFICATION NO.	APPLICABLE FOR B W	D W	D C	A W	A C	SPECIFICATION	RESPONSE
CS 211	x					Time taken for the following task, which is the complement of CS 209: A 4-bit operand has been produced by arithmetic operations and stored for continued computational use. What is the time needed to store it in the middle of a computer word without changing the contents of the rest of the word? (If CS 209 was zero, probably this will be zero also.)	μsec.
CS 212		x		x		Time taken for the following task, which is the complement of CS 210: A one-digit operand has been produced by arithmetic operations and stored for continued computational use. What is the time needed to store it in the middle of a computer word without changing the contents of the rest of the word? (If CS 210 was zero, probably this will be zero also.)	μsec.
CS 213	x	x	x	x	x	Time taken to increment and test an index register.	μsec.
CS 214	x	x	x	x	x	Time taken to move an instruction from one part of the main memory to another location.	μsec.
CS 215	x	x	x	x	x	Time taken to move a record of N characters from one part of the main memory to another. (N is to be considered a large number.)	μsec.
CS 216	x	x	x	x	x	Time taken to indirectly address an operand. (If there is no indirect addressing capability, enter "α".)	μsec.

COMPUTER SPECIFICATIONS

PART 2 - PROCESSOR TIMES (CONT'D)

SPECIFICATION NO.	APPLICABLE FOR B W	D W	D C	A W	A C	SPECIFICATION	RESPONSE
CS 217						General table search task* with minimized programming, using sequential search.	μsec.
CS 218						General table search task* with minimized object time, using sequential search method.	μsec.
CS 219						General table search task* with minimized programming, using binary search method.	μsec.
CS 220						General table search task* with minimized object time, using binary search method.	μsec.

* General table search task: Examine a table stored in main memory to find an argument identical with a test argument. The desired answer is the time per argument tested, with initialization time ignored. Arguments are 8 decimal digits long, and arranged in ascending sequence with variable increments between the values of consecutive arguments.

- Sequential search method: The table arguments are examined in straightforward sequential fashion, allowing the automatic table look-up facilities to be used in many computers.
- Binary search method: Assume the table has N entries, where N is 2 raised to any integral power (e.g., 64). First compare the (N/2)th table argument with the test argument. Depending upon the results, examine next the (N/4)th or (3N/4)th argument; then the (N/8)th, (3N/8)th, (5N/8)th, or (7N/8)th argument; etc.

COMPUTER SPECIFICATIONS

PART 2 - PROCESSOR TIMES (CONT'D)

SPECIFI-CATION NO.	APPLICABLE FOR					SPECIFICATION	RESPONSE
	BW	DW	DC	AW	AC		
CS 221						Time required to respond to hardware interrupt condition not due to hardware malfunction, and transfer control to program execution mode.	μsec.
CS 222						Time required to respond to interrupt due to hardware malfunction, take whatever corrective action is possible, and transfer control to program execution.	μsec.
CS 223						Time required to respond to priority job interrupt conditions and transfer control to program execution.	μsec.
CS 224						Time required to transfer control to alternate hardware processor.	μsec.
CS 225						Main memory requirements for resident operating system.	words/char

COMPUTER SPECIFICATIONS

PART 3 - INPUT/OUTPUT TIMES

SPECIFICATION NO.	APPLICABLE FOR B W	D W	D C	A W	A C	SPECIFICATION	RESPONSE
CS 301	x	x		x		General input editing task* with programming minimized and 11-character alphabetic field. Input field is synchronized (i.e., aligned in accordance with computer word structure).	μsec.
CS 302	x	x		x		General input editing task* with programming minimized and 5-digit numeric field. Input field is synchronized.	μsec.
CS 303	x	x		x		General input editing task* with object time minimized and 11-character alphabetic field. Input field is synchronized.	μsec.
CS 304	x	x		x		General input editing task* with object time minimized and 5-digit numeric field. Input field is synchronized.	μsec.
CS 305	x	x		x		General input editing task* with programming minimized, and 11-character alphabetic field. Input field is not synchronized (i.e., it overlaps computer word boundaries).	μsec.
CS 306	x	x		x		General input editing task* with programming minimized, and 5-digit numeric field. Input field is not synchronized.	μsec.

* General input editing task: Take a field stored in main memory in punched card code; verify the legality of the punching; translate as needed; and unpack so that the field can be used directly as an arithmetic operand. The times are differentiated into coding with minimized programming effort or minimized object time; alphabetic or numeric fields; and (for fixed word systems only) input fields synchronized or not synchronized with the computer's word structure. (Where radix conversion is required between card code and computational representation, the conversion time should be included unless the radix conversion can be more efficiently performed off-line. In the latter case, please describe the equipment and procedure to be used for the off-line radix conversion.)

COMPUTER SPECIFICATIONS

PART 3 - INPUT/OUTPUT TIMES (CONT'D)

SPECIFICATION NO.	APPLICABLE FOR B W	D W	D C	A W	A C	SPECIFICATION	RESPONSE
CS 307	x	x		x		General input editing task* with object time minimized and 11-character alphabetic field. Input field is not synchronized.	μsec.
CS 308	x	x		x		General input editing task* with object time minimized and 5-digit numeric field. Input field is not synchronized.	μsec.
CS 309			x		x	General input editing task* with programming minimized and 11-character alphabetic field.	μsec.
CS 310			x		x	General input editing task* with programming minimized and 5-digit numeric field.	μsec.
CS 311			x		x	General input editing task* with object time minimized and 11-character alphabetic field.	μsec.
CS 312			x		x	General input editing task* with object time minimized and 5-digit numeric field.	μsec.

COMPUTER SPECIFICATIONS

PART 3 - INPUT/OUTPUT TIMES (CONT'D)

SPECIFICATION NO.	APPLICABLE FOR A W	D W	D C	A W	A C	SPECIFICATION	RESPONSE
CS 313	x	x		x		General output editing task** with programming minimized and an 11-character alphabetic field. Output field is synchronized.	μsec.
CS 314	x	x		x		General output editing task** with programming minimized and a 6-character numeric field and commercial editing. Output field is synchronized.	μsec.
CS 315	x	x		x		General output editing task** with programming minimized and a 6-character numeric field and scientific editing. Output field is synchronized.	μsec.

** General output editing task: Take a field stored in main memory, insert editing symbols, translate to printer code as needed, and move an output area in main memory. The times are differentiated into coding with minimized programming effort or minimized object time; alphabetic, commercial numeric, or scientific numeric fields (see below); and (for fixed word systems only) output fields synchronized or not synchronized with the computer's word structure.

- Alphabetic field: The stored field is simply moved to the output area, with translation to printer code if needed.

- Commercial editing on numeric field: The stored field is in cents. Insert floating dollar sign, comma, and decimal point. Place CR or DB alongside, depending upon the sign.

- Scientific editing on numeric field: The stored field requires zero suppression and insertion of a sign and decimal point, with two decimal places to the right of the point.

(Where numeric fields require radix conversion between the computational representation and the printer code, the conversion time should be included <u>unless</u> the radix conversion can be more efficiently performed off-line. In the latter case, please describe the equipment and procedure to be used for the off-line radix conversion.)

COMPUTER SPECIFICATIONS

PART 3 - INPUT/OUTPUT TIMES (CONT'D)

SPECIFI-CATION NO.	APPLICABLE FOR BW	DW	DC	AW	AC	SPECIFICATION	RESPONSE
CS 316	x	x		x		General output editing task** with object time minimized and an 11-character alphabetic field. Output field is synchronized.	μsec.
CS 317	x	x		x		General output editing task** with object time minimized and a 6-character numeric field and commercial editing. Output field is synchronized.	μsec.
CS 318	x	x		x		General output editing task** with object time minimized and a 6-character numeric field and scientific editing. Output field is synchronized.	μsec.
CS 319	x	x		x		General output editing task** with programming minimized and an 11-character alphabetic field. Output field is not synchronized.	μsec.
CS 320	x	x		x		General output editing task** with programming minimized and a 6-character numeric field and commercial editing. Output field is not synchronized.	μsec.
CS 321	x	x		x		General output editing task** with programming minimized and 6-character numeric field and scientific field and scientific editing. Output field is not synchronized.	μsec.
CS 322	x	x		x		General output editing task** with object time minimized and an 11-character alphabetic field. Output field is not synchronized.	μsec.
CS 323	x	x		x		General output editing task** with object time minimized and a 6-character numeric field and commercial editing. Output field is not synchronized.	μsec.

COMPUTER SPECIFICATIONS

PART 3 - INPUT/OUTPUT TIMES (CONT'D)

SPECIFICATION NO.	APPLICABLE FOR BW	DW	DC	AW	AC	SPECIFICATION	RESPONSE
CS 324	x	x		x		General output editing task** with object time minimized and a 6-character numeric field and scientific editing. Output field is not synchronized.	μsec.
CS 325			x		x	General output editing task** with programming minimized and an 11-character alphabetic field.	μsec.
CS 326			x		x	General output editing task** with programming minimized and a 6-character numeric field and commercial editing.	μsec.
CS 327			x		x	General output editing task** with programming minimized and a 6-character numeric field and scientific editing.	μsec.
CS 328			x		x	General output editing task** with object time minimized and an 11-character alphabetic field.	μsec.
CS 329			x		x	General output editing task** with object time minimized and a 6-character numeric field and commercial editing.	μsec.
CS 330			x		x	General output editing task** with object time minimized and a 6-character numeric field and scientific editing.	μsec.

COMPUTER SPECIFICATIONS

PART 4 - MAGNETIC TAPE

SPECIFI-CATION NO.	SPECIFICATION	RESPONSE
CS 401	Number of magnetic tapes which can be reading with processing proceeding.	
CS 402	Number of magnetic tapes which can be reading with no processing proceeding.	
CS 403	Number of magnetic tapes which can be writing with processing proceeding.	
CS 404	Number of magnetic tapes which can be writing with no processing proceeding.	
CS 405	Total number of magnetic tapes which can be reading and/or writing with processing proceeding.	
CS 406	Total number of magnetic tapes which can be reading and/or writing with no processing proceeding.	
CS 407	Can more than one program be running at one time? (Yes or No)	
CS 408	Peak speed, in alphanumeric characters per second.	char/sec.
CS 409	Cost in characters of a tape gap when passed over as quickly as possible. *	chars.
CS 410	Minimum block length, in alphanumeric characters.	chars.
CS 411	Block length increment, in alphanumeric characters.	chars.
CS 412	Maximum block length, in alphanumeric characters.	chars.

* Can be determined by multiplying the minimum time to cross the inter-block gap, in seconds, by the peak data transfer rate, in characters per second.

COMPUTER SPECIFICATIONS

PART 4 - MAGNETIC TAPE (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 413	Central processor time used per alphanumeric character read or written.	μsec.
CS 414	Number of decimal digits per alphanumeric character in the computer's internal code.	
CS 415	Number of decimal digits per alphanumeric character in the magnetic tape code.	
CS 416	Number of alphanumeric characters per computer word.	
CS 417	Maximum blocking factor for card image input available using standard routines.	
CS 418	Maximum blocking factor for line images output available using standard routines.	
CS 419	Additional central processor time used per alphanumeric character when scatter-read gather-write facilities are used. (If such facilities are not available, write N. A.).	μsec.

COMPUTER SPECIFICATIONS

PART 5 - LINE PRINTERS

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
CS 501	Skip speed(s)	inches/sec.
CS 502	Effective printer speed* for alphanumeric character set (letters A-Z; numerals 0-9, and 4 special symbols) at interline space: 1/2 inch 1 inch 2 inches 3 inches 4 inches 5 inches 6 inches	 lpm lpm lpm lpm lpm lpm lpm
CS 503	Effective printer speed for numeric character set (numerals 0-9 and 4 special symbols) at interline space: 1/2 inch 1 inch 2 inches 3 inches 4 inches 5 inches 6 inches	 lpm lpm lpm lpm lpm lpm lpm
CS 504	Print width of printed page	char
CS 505	Channel load time** per line printed	msec
CS 506	Processor load per line printed	msec

* Including overheads such as start-stop time, and any others peculiar to the device in question.

** Channel Load Time is the time during which an I/O data channel is exclusively occupied while handling a unit (e.g., bit, character, word or block) data transfer.

COMPUTER SPECIFICATIONS

PART 6 - CARD EQUIPMENT

SPECI-FICATION NO.	SPECIFICATION		RESPONSE
CS 601	State effective card reader speed, channel load time*, and processor usage including confirmation of image under demand conditions.		
	CARD READER SPEED (Rightmost column read)	10 20 30 40 50 60 70 80	Cards per minute
	CHANNEL LOAD TIME (Rightmost column read)	10 20 30 40 50 60 70 80	μsec.
	PROCESSOR USAGE (Rightmost column read)	10 20 30 40 50 60 70 80	μsec.

* Channel Load Time is the time during which an I/O data channel is exclusively occupied while handling a unit (e.g. bit, character, word or block) data transfer.

COMPUTER SPECIFICATIONS

PART 6 - CARD EQUIPMENT (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 602	State effective card reader speed, channel load time, and processor usage including confirmation of image under conditions of continuous read. CARD READER SPEED (Rightmost column read) 10 20 30 40 50 60 70 80 CHANNEL LOAD TIME (Rightmost column read) 10 20 30 40 50 60 70 80 PROCESSOR USAGE (Rightmost column read) 10 20 30 40 50 60 70 80	Cards per minute μsec. μsec.

* Channel Load Time is the time during which an I/O data channel is exclusively occupied while handling a unit (e.g. bit, character, word or block) data transfer.

COMPUTER SPECIFICATIONS

PART 6 - CARD EQUIPMENT (CONT'D)

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
CS 603	For card reader ________________: State time within which the next instruction must be given (executive) to avoid missing clutch points.	msec.
CS 604	For card reader ________________: State time between clutch points.	msec.
CS 605	State effective card punch speed, channel load time* and processor usage including confirmation of image under demand conditions. CARD PUNCH SPEED (Rightmost column read) 10, 20, 30, 40, 50, 60, 70, 80 CHANNEL LOAD TIME (Rightmost column read) 10, 20, 30, 40, 50, 60, 70, 80 PROCESSOR USAGE (Rightmost column read) 10, 20, 30, 40, 50, 60, 70, 80	Cards per minute μsec. μsec.

* Channel Load Time is the time during which an I/O data channel is exclusively occupied while handling a unit (e.g. bit, character, word, or block) data transfer.

PART 6 - CARD EQUIPMENT (CONT'D)

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
CS 606	State effective card punch speed, channel load time*, and processor usage including confirmation of image under demand conditions of continuous punching. CARD PUNCH SPEED (Rightmost column read) 10, 20, 30, 40, 50, 60, 70, 80	Cards per minute
	CHANNEL USAGE (Rightmost column read) 10, 20, 30, 40, 50, 60, 70, 80	µsec.
	PROCESSOR USAGE (Rightmost column read) 10, 20, 30, 40, 50, 60, 70, 80	µsec.
CS 607	For card punch______________: State time within which the next instruction must be given to avoid missing clutch points.	msec.
CS 608	For card punch______________: State time in between clutch points.	msec.

* Channel Load Time is the time during which an I/O data channel is exclusively occupied while handling a unit (e.g. bit, character, word or block) data transfer.

COMPUTER SPECIFICATIONS

PART 7 - RANDOM ACCESS DEVICES

<table>
<tr><th>SPECIFICATION NO.</th><th>SPECIFICATION</th><th>RESPONSE</th></tr>
<tr><td>CS 701</td><td>Effective access times[1] and channel load times[2] in microseconds for random access device. Fill out the tables below for the following conditions: Random records[3], known machine address[4], throughput maximized, and minimum programming[5].

EFFECTIVE ACCESS TIMES
<table>
<tr><th>Record Size \ File Size Char.(6)</th><th>10³</th><th>10⁴</th><th>10⁵</th><th>10⁶</th><th>10⁷</th><th>10⁸</th></tr>
<tr><td>100 char.</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>500 char.</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1000 char.</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1500 char.</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

CHANNEL LOAD TIMES
<table>
<tr><th>Record Size \ File Size Char.</th><th>10³</th><th>10⁴</th><th>10⁵</th><th>10⁶</th><th>10⁷</th><th>10⁸</th></tr>
<tr><td>100 char.</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>500 char.</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1000 char.</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1500 char.</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td><td></td></tr>
</table>

[1] Effective Access Time is the time interval between the instant the user's program calls for a transfer of data to or from the store or input/output unit and the instant the operation is completed. Thus, effective access time is the sum of the transfer time and waiting time, including latency time, e.g., time to do table look-ups, etc., as required.

[2] Channel Load Time is the time during which an I/O data channel is exclusively occupied while handling a unit (e.g., bit, character, word, or block) data transfer.

[3] Random Record is a record stored in a file where the sequence of the record in question (i.e., the random record) may or may not be related to the sequence of the adjacent records. It should be noted that this is a characteristic of a specific file and a specific pattern of transactions, not a specific file alone.

[4] Known Machine Address refers to a physical address which defines the actual position of a record in an on-line file.

[5] Minimum Programming means the use whenever possible of generalized standard software routines and/or packages.

[6] Note that the file size is given in terms of the data file, not system capacity.

PART 7 - RANDOM ACCESS DEVICES (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 702	Effective access times and channel load times in microseconds for random access device. Fill out the tables below for the following conditions: Random records, known machine address, throughput maximized and specialized programming as needed.[7]	

EFFECTIVE ACCESS TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

CHANNEL LOAD TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

[7] Specialized Programming means a sub-routine especially prepared for the user's program with preliminary knowledge of which part of the program is the limiting factor.

PART 7 - RANDOM ACCESS DEVICES (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 703	Effective access times and channel load times in microseconds for random access device. Fill out the tables below for the following conditions: Random records, known machine address, response time minimized, and minimum programming.	

EFFECTIVE ACCESS TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

CHANNEL LOAD TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^8	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 704	Effective access times and channel load times in microseconds for random access device. Fill out the tables below for the following conditions: Random records, known machine address, response time minimized and specialized programming as necessary.	

EFFECTIVE ACCESS TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

CHANNEL LOAD TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

PART 7 - RANDOM ACCESS DEVICES (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 705	Effective access times and channel load times in microseconds for random access device. Fill out the tables below for the following conditions: Random records, machine address unknown,[8] throughput maximized, and minimum programming.	

EFFECTIVE ACCESS TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

CHANNEL LOAD TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

[8] Unknown Machine Address refers to a sufficient definition of the identification of a record so that its known machine address can be traced (via links) looked up or computed.

PART 7 - RANDOM ACCESS DEVICES (CONT'D)

SPECIFI-CATION NO.	SPECIFICATION	RESPONSE
CS 706	Effective access times and channel load times in microseconds for random access device. Fill out the tables below for the following conditions: Random records, machine address unknown, throughput maximized, and specialized programming as necessary.	

EFFECTIVE ACCESS TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

CHANNEL LOAD TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

PART 7 - RANDOM ACCESS DEVICES (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 707	Effective access times and channel load times in microseconds for random access device. Fill out the tables below for the following conditions: Random records, machine address unknown, response time minimized, and minimum programming.	

EFFECTIVE ACCESS TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

CHANNEL LOAD TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

COMPUTER SPECIFICATIONS

PART 7 - RANDOM ACCESS DEVICES (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 708	Effective access times and channel load times in microseconds for random access device. Fill out the tables below for the following conditions: Random record, machine address unknown, response time minimized, and specialized programming as necessary..	

EFFECTIVE ACCESS TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

CHANNEL LOAD TIMES

Record Size \ File Size Char.	10^3	10^4	10^5	10^6	10^7	10^8
100 char.						
500 char.						
1000 char.						
1500 char.						

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 709	Maximum number of characters which can be transferred on a single request (if less than 2,000)	char.
CS 710	Effective data transfer rate	char./sec.

COMPUTER SPECIFICATIONS

PART 7 - RANDOM ACCESS DEVICES (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
CS 711	Is the file physically removable from the file drive (e.g., portable discs, cartridges, etc.)?	yes/no
CS 712	If CS 711 is yes, state the time involved in changing the file unit. (i.e., time required to unload a unit and load a new one.)	

APPENDIX II

EXTENDED MACHINE SPECIFICATIONS

DIRECTIONS FOR COMPLETING EXTENDED MACHINE SPECIFICATIONS

The extended machine specifications are intended to provide information as to how the software interacts with the hardware configuration, and as a result provide a measurement of the effect of the software upon the hardware performance.

The combined ratios of the hardware performance times (derived from hand-tailored coding) for the given macrofunctions to the extended machine performance times for the same macrofunctions represent the relative efficiency of the selected software system for a given machine configuration. It is intended that each specification be programmed in the source language of the software system being evaluated and then executed on the target computer. The response, then will be the execution time required to execute the code produced by the source language statement(s) after compilation or assembly, whichever is applicable. Manufacturers should be prepared to supply documentation of the object program macroloop coding. The specifications in this section refer to their corollaries in the computer specification.

Specifications ES116 through ES119 deal with general overhead problems of operating or executive systems. The response should be derived from actual operating conditions where this information is available. If no operating system is used in the installation, or if there is no operating system available for the particular computer, mark the response "N/A" (not applicable).

In the case of a proposed or only partially completed software system, give estimates of timing information, and mark the response as an estimate.

A separate set of specifications for the extended machine (hardware/software interaction) should be completed for each software system used (or to be used) in conjunction with a particular computer system.

EXTENDED MACHINE

(HARDWARE/SOFTWARE INTERACTION) SPECIFICATIONS

SPECIFICATION NO.	TYPE[1]	SPECIFICATION	RESPONSE
ES 101	C/A	Add two operands in main memory and store the sum. (A+B → C). A and B > 4 digits. Refer to CS 201 for hardware corollary.	μsec.
ES 102	C/A	Multiply two operands and store the product. (AxB → C), A and B > 4 digits. Refer to CS 202 for hardware corollary.	μsec.
ES 103	C/A	Divide an X digit operand by a Y digit operand and store the quotient. (X÷Y →Y), X and Y > 4 digits. Refer to CS 203 for hardware corollary.	μsec.
ES 104	C/A	Time required to index an operand. Refer to CS 206 for hardware corollary.	μsec.
ES 105	C/A	Compare two operands (each operand eight decimal digits or equivalent) and transfer control to one of two arbitrary locations, based on the result of the comparison (e.g., IF A GEQ B THEN GO TO α). Refer to CS 208 for hardware corollary.	μsec.
ES 106	C/A	Transfer control on a 6-position program switch where the switching variables are one-digit operands having the values 1, 2, 3, 4, 5, or 6. Include the time required to check the value and transfer control (e.g., SWITCH ALPHA (A, B, C, D, E, F). Refer to CS 208 for hardware corollary.	μsec.
ES 107	C/A	Move a record of N characters from one part of main memory to another. (Assume N to be a large number.) Refer to CS 212 for hardware corollary.	μsec.

[1] C - Compiler
A - Assembler
O - Operating System

EXTENDED MACHINE

(HARDWARE/SOFTWARE INTERACTION) SPECIFICATIONS (CONT'D)

SPECIFI-CATION NO.	TYPE	SPECIFICATION	RESPONSE
ES 108	C/A	General input editing task[2] for 11-character alphabetic field with object time minimized. (Applicable for decimal or alphanumeric character-oriented machine.) Refer to CS 311 for hardware corollary.	μsec.
ES 109	C/A	General input editing task[2] for 5-character numeric field with object time minimized. (Applicable for decimal or alphanumeric character-oriented machines.) Refer to CS 312 for hardware corollary.	μsec.

[2] General input editing task: Take a field stored in main memory in punched card code; verify the legality of the punching; translate as needed; and unpack so that the field can be used directly as an arithmetic operand. The times are differentiated into coding with minimized programming effort or minimized object time; alphabetic or numeric fields; and (for fixed word systems only) input fields synchronized or not synchronized with the computer's word structure. (Where radix conversion is required between card code and computational representation, the conversion time should be included unless the radix conversion can be more efficiently performed off-line. In the latter case, please describe the equipment and procedure to be used for the off-line radix conversion.)

EXTENDED MACHINE

(HARDWARE/SOFTWARE INTERACTION) SPECIFICATIONS (CONT'D)

SPECIFI-CATION NO.	TYPE	SPECIFICATION	RESPONSE
ES 110	C/A	General output editing task[3] for an 11-character alphabetic field with programming minimized. (Applicable for decimal or alphanumeric character-oriented machines.) Refer to CS 325 for hardware corollary.	μsec.
ES 111	C/A	General output editing task[3] for a 6-character numeric field using commercial editing with programming minimized. (Applicable for decimal or alphanumeric character-oriented machines.) Refer to CS 326 for hardware corollary.	μsec.

---

[3] General output editing task: Take a field stored in main memory, insert editing symbols, translate to printer code as needed, and move to an output area in main memory. The times are differentiated into coding with minimized programming effort or minimized object time; alphabetic, commercial numeric, or scientific numeric fields (see below); and (for fixed word systems only) output fields synchronized or not synchronized with the computer's word structure.

- Alphabetic field: The stored field is simply moved to the output area, with translation to printer code if needed.
- Commercial editing on numeric field: The stored field is in cents. Insert floating dollar sign, comma, and decimal point. Place CR or DB alongside, depending upon the sign.
- Scientific editing on numeric field: The stored field requires zero suppression and insertion of a sign and decimal point, with two decimal places to the right of the point.

(Where numeric fields require radix conversion between the computational representation and the printer code, the conversion time should be included unless the radix conversion can be more efficiently performed off-line. In the latter case, please describe the equipment and procedure to be used for the off-line radix conversion.)

EXTENDED MACHINE

(HARDWARE/SOFTWARE INTERACTION) SPECIFICATIONS (CONT'D)

SPECIFI-CATION NO.	TYPE	SPECIFICATION	RESPONSE
ES 112	C/A	General output editing task[4] for a 6-character numeric field using scientific editing with programming minimized. (Applicable for decimal or alphanumeric character-oriented machines.) Refer to CS 327 for hardware corollary.	μsec.
ES 113	C/A	General table search task[4] using sequential search with programming minimized. Refer to CS 217 for hardware corollary.	μsec.
ES 114	C/A	General table search task[4] using binary search with programming minimized. Refer to CS 218 for hardware corollary.	μsec.
ES 115	O	Time required to respond to hardware interrupt condition not due to hardware malfunction and transfer control back to users' program. Refer to CS 221 for hardware corollary.	μsec.
ES 116	O	Time required to respond to interrupt due to hardware malfunction, take whatever corrective action is possible and transfer control to users' program. Refer to CS 222 for hardware corollary.	μsec.

[4] General table search task: Examine a table stored in main memory to find an argument identical with a test argument. The desired answer is the time per argument tested, with initialization time ignored. Arguments are 8 decimal digits long, and arranged in ascending sequence with variable increments between the values of consecutive arguments.

- Sequential search method: The table arguments are examined in straightforward sequential fashion, allowing the automatic table look-up facilities to be used in many computers.
- Binary search method: Assume the table has N entries, where N is 2 raised to any integral power (e.g., 64). First compare the (N/2)th table argument with the test argument. Depending upon the results, examine next the (N/4)th or (3N/4)th argument; then the (N/8)th, (3N/8)th, (5N/8)th, or (7N/8)th argument; etc.

EXTENDED MACHINE

(HARDWARE/SOFTWARE INTERACTION) SPECIFICATIONS (CONT'D)

SPECIFICATION NO.	TYPE	SPECIFICATION	RESPONSE
ES 117	O	Time required to respond to priority job interrupt conditions (hardware or software) and transfer control to priority users' program. Refer to CS 223 for hardware corollary.	μsec.
ES 118	O	Time required to transfer control to alternate hardware processor. Refer to CS 224 for hardware corollary.	μsec.
ES 119	O	Main memory requirements for resident operating system. Refer to CS 225 for hardware corollary.	words/ char.

APPENDIX III

PROBLEM SPECIFICATIONS

DIRECTIONS FOR COMPLETING PROBLEM SPECIFICATIONS

The problem specifications are to be completed after the appropriate application class has been determined through the task classification analysis. This set of specifications pertains to both classes of file processing applications, static and dynamic. The specifications are written in such a manner as to lend themselves to most applications, and the responses define the processing operations required in the automated solution of the problem. Since these specifications are used as part of a process to determine the effectiveness of the total ADP installation, they must be completed as accurately and objectively as possible. The problem specifications are divided into eight distinct, but not necessarily exclusive parts.

Part 1 is general information which indicates whether the processing will be accomplished using random access storage or a magnetic tape subsystem. If the application is to be accomplished using random access equipment, some additional specifications are required.

Parts 2 and 3 are designed to furnish details relative to the transactions or messages to be processed.

The Transaction File contains all the new information which is to be used to update the Master File, create reports, etc. In general, the information needed relative to the transaction data for the purpose of estimating computer performance is related to three factors:

(1) The sequence in which the transactions are available.

(2) The description of each type of transaction.

(3) The work involved in processing each type of transaction.

The specification for (1) can be separated from the others, and is included on a separate sheet. The details of each of the specification queries is intended to be self-explantory. The following pages contain detailed examples of how the individual items are to be completed. One sheet is to be completed for each transaction record type. On each sheet, details are first sought regarding the physical appearance of the file, how many, of what size records, etc.

Next, more complicated specifications are requested. These deal with the estimates of selected programming steps required to process each transaction type. The workload is divided into three types of work, arbitrarily called "Simple Update," "Complex Update," and "Table Reference." These have been chosen because it is not valid to estimate the number of multiplication executions a process will require per transaction from only a knowledge of the number of fields to be involved in the processing. Similarly, the computer's Add/Multiply or Add/Table Reference ratios vary, so that distinction must be made between these types of operations. In this respect, they differ from operations such as subtractions or divisions, which can be regarded as equivalent to additions or multiplications respectively without imperiling the accuracy of the results. The choice of these three processing categories represents a compromise between the minimum required data and normal methods for approximating the instruction mix (e.g., the Gibson Mix) that lead to inaccuracies. It is felt that a systems analyst does not have a detailed knowledge of the number of arithmetic or logical operations to be performed (including loop iteration). Therefore, the provision of categories of operation types is also intended to make it easier to indicate relative degrees of varying processing complexity levels.

Due to the absence of absolute programming specifications for any problem, some degree of error in calculation of estimated running time is permissible. The main problem has been to avoid the introduction of typical or average numbers for cases where individual problem and machine conditions demand actions that are significantly different from those governing general operating conditions. Examples of such actions might be the packing of magnetic tape formats through use of binary instead of decimal number representation, the loose packing (format arrangement) of magnetic tape records in order to eliminate editing within the central processor, the possible dynamic variation in record sizes of magnetic tape master file records, etc.

In considering the individual specifications which are required for each record type (separate record types for all files are specified on separate pages), it is unlikely that all the records of the specific type go through the same process. Some will go through Process A, and some through Process B as well. To allow for this there is space in FS370-390 for four separate "processes" to be described. For instance, if in processing an inventory receipt transaction it is required to price the value of the merchandise in some cases, but not in others, such a situation might arise where some goods are received on consignment as opposed to those which are purchased. This would indicate that each day 10,000 issue vouchers were to be processed; 9,500 of them would involve only two addition, subtraction, or comparison operations, and 500 would involve three additions and a multiplication or division.

It will be noted that Specifications PS351 and PS361 deal with the use of random access equipment, and should be marked "N/A"(not applicable) in a process which uses magnetic tape exclusively.

Parts 4 and 5 are specifications which, when completed, define the master file. The Master File is described in a manner similar to the Transcription File; that is, one general sheet plus one Master File Record Sheet for each type of few simple queries, which are self-explanatory. The major details are included in the individual Master File Record Sheets.

These Record Sheets are designed to reflect the composition of the file. Thus, the first three specifications include the number of records, the average number of characters per record, and the average number of numeric digits per record. While the number of records in the Master File certainly varies from day to day, there is an average number. This is the value to be entered in the specification. Similarly, while there may be considerable variation in the size of individual records (relative to variable length record files), there should only be one average size on any specific day. In cases where variable record sizes are treated as headers and trailers, each trailer type should be treated as a separate type within the Master File.

After describing the makeup of the Master File Records, the next group of specifications relate to the work induced by each individual record. Very frequently, a Master File record will not induce any work whatsoever, and in these cases no entries need be made here. In such cases, all the processing involved will have been described either under the Transaction or Report Files.

Parts 6 and 7 are specifications for the report file. In Part 6, the general description of the report file is given. Part 7 requests details on each separate report. Hence, one copy of Part 7 must be completed for each separate report produced in the processing cycle. The Report Description Sheet is made out for each report type. It is not normally necessary to break down headings, subtotals, combinations, final totals, etc. The details requested are analogous to those in the previous sections.

The report record size, number of characters that must be computed (i.e., edited for each line), the percentage of these that are alpha, its medium (in this case restricted to magnetic tape), and whether the print line format can be varied to suit the editing properties of the computer concerned are requested. This information is designed to indicate the degree of internal processing that must be performed. It is assumed that there will be a printed line produced for each Master File Record that is updated, unless otherwise indicated. If any information on the size of a preprinted form (where used) is available, it should be noted in the comments applicable to this section.

Note specification PS730, which requests a statement of the report medium (or media). The term "Hard Copy" implies the use of at least one printer. Therefore, Part 8 of the Problem Specifications must be completed.

Part 8 contains specifications for printed output to be accomplished via a line printer. A copy of Part 8 should be completed for each different printed report required in the problem solution.

PROBLEM SPECIFICATIONS

PART 1 - GENERAL

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
PS 101	Is the application suited to magnetic tape oriented processing?	yes/no
PS 102	Is the application suited to random access processing?	yes/no
PS 103	If this is a random access application, state objective as either (a) maximize number of records processed or (b) minimize access time to a specific record.	

PROBLEM SPECIFICATIONS

PART 2 - TRANSACTION FILE

(Use 1 per Transaction File)

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
PS 201	No. of transaction records per cycle (standard).	
PS 202	No. of transaction records per cycle (peak).	
PS 203	Will the transactions be sorted in main file order?	yes/no
PS 204	Will the transactions already be on magnetic tape?	yes/no
PS 205	Will the transactions already be on the random access device?	yes/no
PS 206	May the analyst alter the format of the transaction records to suit the particular system?	yes/no
PS 207	No. of transaction types in the file.	

(Describe each type individually on a separate Transaction File Record Type Sheet.)

PROBLEM SPECIFICATIONS

PART 3 - TRANSACTION RECORD DETAILS

(Use 1 per Record Type)

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
PS 310	No. of records of this type in the Transaction File.	
PS 320	No. of characters (including alphabetic, numeric, and special characters) per record.	
PS 330	No. of numeric digits per record. *	
PS 340	Average number of active fields per records (an active field is one which is used or referred to during processing). **	

*Assumed to be equal to (PS 320) ÷ 2, if not given.
**Assumed to be equal to PS 310, if not given.

DETAILS OF EACH SIGNIFICANT WORK PATH

In this section, each process which results from this transaction type is enumerated, and details are given to show how frequently the process is executed. The volume of processing which takes place during each execution of the process is described in terms of "Simple Update or Equivalent Operations," "Complex Update Steps or Equivalent Operations," and "Table References."

	PROCESS NAME:	
PS 350	% of records using this work path.	
PS 360	No. of Simple Field Updates or Equivalent Operations per record. (This is the equivalent of the sum of the add/subtract and comparison operations needed to process a record.)	
PS 361	State the number of files accessed per transaction: read, write, or write and check usages of the random access records if necessary during the processing.	
PS 362	State the number of read, read/write, and read/write/check usages of the random access records during the processing.	

PROBLEM SPECIFICATIONS

PART 3 - TRANSACTION RECORD DETAILS (CONT'D)

	PROCESS NAME:								
PS 370	No. of Complex Field Update Steps or Equivalent Operations per record. (This is the equivalent of the sum of multiply and divide operations per record.)								
PS 380	Average no. of decimal digits in the numeric operands used during the process.								
PS 390	No. of associated values (A) extracted from tables in execution of the work path processing, arranged by table size involved (T). (Ignore T if tables have less than 50 entries.)	A	T	A	T	A	T	A	T

PROBLEM SPECIFICATIONS

PART 4 - MASTER FILE

(Use 1 per Master File)

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
PS 401	No. of records in the master file.	
PS 402	No. of major record types in the file.	
PS 403	What are the time intervals at which random records must be updated to remain usable?	

(Describe each type individually on a separate Master File Record Type sheet.)

PROBLEM SPECIFICATIONS

PART 5 - MASTER FILE RECORD DETAILS

(Use 1 per Record Type)

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
PS 510	No. of records of this type in the Master File.	
PS 520	No. of characters (including alphabetic, numeric, and special characters) per record.	
PS 530	No. of numeric digits per record. *	
PS 540	Average number of active fields per record (an active field is one which is used or referred to during processing). **	

*Assumed to be equal to (PS 520) ÷ 2, if not given.
**Assumed to be equal to PS 510, if not given.

DETAILS OF EACH SIGNIFICANT WORK PATH

In this section, each process which results from this record type is enumerated, and details are given to show how frequently the process is executed. The volume of processing which takes place during each execution of the process is described in terms of "Simple Update or Equivalent Operations." "Complex Update Steps or Equivalent Operations." and "Table References."

	PROCESS NAME								
PS 550	% of Records using this work path.								
PS 560	No. of Simple Field Updates or Equivalent Operations per record. (This is the equivalent of the sum of the add/subtract and comparison operations needed to process a record.)								
PS 570	No. of Complex Field Update Steps or Equivalent Operations per record. (This is the equivalent of the sum of multiply and divide operations per record.								
PS 580	Average no. of decimal digits in the numeric operands used during the process.								
PS 590	No. of associated values (A) extracted from tables in execution of the work path processing, arranged by table size involved (T). (Ignore T if tables have less than 50 entries.)	A	T	A	T	A	T	A	T

PROBLEM SPECIFICATIONS

PART 6 - REPORT FILE

(Use 1 per Report File)

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
PS 601	No. of report records per cycle (standard).	
PS 602	No. of report records per cycle (peak).	
PS 603	Should the reports be sorted in main file order?	yes/no
PS 604	May the reports be placed on magnetic tape for off-line printing?	yes/no
PS 605	May the analyst alter the format of the report records to suit the particular system?	yes/no
PS 606	No. of report types in the file.	
PS 607	Are logical records to be placed on separate cards?	yes/no

(Describe each type individually on a separate Report File Record Type Sheet.)

PROBLEM SPECIFICATIONS

PART 7 - REPORT FILE RECORD DETAILS

(Use 1 per Record Type)

SPECI-FICATION NO.	SPECIFICATION	RESPONSE
PS 710	No. of records of this type in the Report File.	
PS 720	No. of characters (including alphabetic, numeric, and special characters) per record.	
PS 730	Report Media (Communication line, CRT display, Hard Copy, etc.)	
PS 731	No. of printed lines per record.	
PS 740	Average number of active alphabetic fields per record. (An active field is one which is prepared or edited during processing rather than simply copied from some other document.)*	
PS 750	Average number of active numeric fields per record. (An active field is one which is prepared or edited during processing rather than simply copied from some other document.)**	

*Assumed to be equal to (PS 720) ÷ 20, if not given.

**Assumed to be equal to (PS 720) ÷ 10, if not given.

PROBLEM SPECIFICATIONS

PART 8 - PRINTED REPORT RECORD

SPECIFICATION NO.	SPECIFICATION	RESPONSE
PS 810	Width of printed form in number of characters.	
PS 820	No. of printed alphanumeric lines per physical form.	
PS 830	No. of printed numeric lines per physical form.	
PS 840	No. of printed lines per heading.	
PS 850	No. of printed lines in report body.	
PS 860	No. of printed lines per footing.	
PS 870	Is body of report numeric or alphanumeric?	
PS 880	Length of printed form expressed in inches of paper.	

APPENDIX IV

SOFTWARE SPECIFICATIONS

DIRECTIONS FOR COMPLETING SOFTWARE SPECIFICATIONS

The software specifications have been designed to measure those aspects of a software system which contribute to the effective utilization of the programming staff. As the forms indicate, a scoring method is provided to allow the software to be qualitatively evaluated, independently of the individual machine from the viewpoint of providing facilities to aid the programmer who uses the software system. Specifications are presented which query the completeness of the software relative to the task or job which must be programmed using the source language. Also required are specifications which indicate how easy and effectively it can be used by the programmer in creating an operating program.

The specifications and scoring rules are straightforward and most can be answered with a "yes/no" response. Should a particular specification not be applicable in the environment in which the effectiveness evaluation is conducted, the response should be marked "N/A" (not applicable). If a precise response is not indicated in the software documentation, the information can usually be obtained from the manufacturer or other supplier. A response obtained in this way should be so indicated. Answers obtained for projected software systems on the basis of specifications only should be appropriately identified.

Scoring rules for each specification are included with it and the score should be entered when the response is indicated. This avoids complex scoring procedures which would otherwise have to be completed after completing the specification.

A specification form should be completed for each software system either in use or proposed for use in the installation. Whenever it is known that software has been modified by a responsible source, the specification should be updated to reflect the modification.

SOFTWARE SPECIFICATIONS

PART 1 - DIAGNOSTICS

SPECI-FICATION NO.	SPECIFICATION	RESPONSE	SCORE
SS 101	Are program diagnostic facilities available in the source language? (If response is "no," enter score of 0, and omit Specifications SS102-SS104.)	yes/no	
SS 102	Consider three types of user program diagnostic facilities, snapshot, trace, and post-mortem diagnostic. Enter as the response and score the number of these facilities available in the software system under consideration (1, 2, or 3).		
SS 103	State whether 1) User must go to another (usually lower level) language for program checkout. 2) Checkout can be accomplished in the source language. Enter as the response and score the number of the appropriate condition (either 1 or 2)		
SS 104	Can program changes be made with partial recompilation or reassembly? Score 1 for "yes,", 0 for "no."	yes/no	
SS 105	Are there facilities provided to check the syntax* of the source program? Score 1 for "yes," 0 for "no." If the response is "no," omit Specifications SS106-SS111 below.	yes/no	
SS 106	Are the source language diagnostics applicable to only certain types of source language entries (e.g., arithmetic statements)? Score 0 for "yes," 1 for "no."	yes/no	
SS 107	Are the source language diagnostic error messages limited to a single message at the end of compilation or assembly? Score 0 for "yes," 1 for "no."	yes/no	

*The syntax of a language is the set of rules which define a well-formed (correct) source language entry.

SOFTWARE SPECIFICATIONS

PART 1 - DIAGNOSTICS (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE	SCORE
SS 108	Are the source language diagnostic error messages indicative of the type of error correction procedure to be followed? Score 1 for "yes," 0 for "no."	yes/no	
SS 109	Are the source language diagnostics indicative of both the error type and specific error found? Score 3 for "yes." If yes, omit SS110-SS111.	yes/no	
SS 110	Are the source language diagnostics indicative of the specific error? Score 2 for "yes." If yes, omit SS111.	yes/no	
SS 111	Are the source language diagnostics indicative of the type of error found? Score 1 for "yes," 0 for "no."	yes/no	
SS 112	Are there limitations on the types of compilation or assembly errors to which the diagnostics apply? (e.g., exceeding memory capacity, etc.) Score 0 for "yes," 1 for "no."	yes/no	
SS 113	Can the compiler or assembler correct certain types of errors on-line to allow the possible execution of the object program? (e.g., correct from program or statement context improperly expressed operations). Score 1 for "yes," 0 for "no."	yes/no	
SS 114	Can the compiler or assembler recover from certain types of hardware failure and continue processing? Score 1 for "yes," 0 for "no."	yes/no	

SOFTWARE SPECIFICATIONS

PART 2 - PROGRAM STRUCTURING ELEMENTS

SPECI-FICATION NO.	SPECIFICATION	RESPONSE	SCORE
SS 201	How many commonly used programming devices such as iteration, switches, etc., have been implemented in the compiler or assembler via simple statements in the language? (e.g., FOR, PERFORM, SWITCH, JUMP.) Score 0 for none, 1 for less than five, and 2 for more than five such devices.		
SS 202	How many levels of subroutine nesting are allowed in the compiler or assembler? Score 0 for none, 1 for less than three levels, 2 for four to nine levels, and 3 for more than nine levels.		
SS 203	Are both open and closed subroutines permitted? If response is "yes," score 2 and omit Specification SS204.	yes/no	
SS 204	Is only one type of either open or closed subroutine permitted? Score 1 for "yes," 0 for "no."	yes/no	

SOFTWARE SPECIFICATIONS

PART 3 - STORAGE ALLOCATION AND PROTECTION FACILITIES

SPECI-FICATION NO.	SPECIFICATION	RESPONSE	SCORE
SS 301	Has dynamic storage allocation facility been included for systems program? Score 1 for "yes," 0 for "no."	yes/no	
SS 302	Has dynamic storage allocation facility been included for user's programs? Score 1 for "yes," 0 for "no."	yes/no	
SS 303	Has dynamic storage allocation facility been included for data blocks? Score 1 for "yes," 0 for "no."	yes/no	
SS 304	Has dynamic storage protection facility been included for systems program? Score 1 for "yes," 0 for "no."	yes/no	
SS 305	Has dynamic storage protection facility been included for user's programs? Score 1 for "yes," 0 for "no."	yes/no	
SS 306	Has dynamic storage protection facility been included for data? Score 1 for "yes," 0 for "no."	yes/no	

SOFTWARE SPECIFICATIONS

PART 4 - PROGRAM LIBRARY

SPECI-FICATION NO.	SPECIFICATION	RESPONSE	SCORE
	Consider a minimum library for a software system to include at least the trigonometric functions for numeric computational application, or SORT/MERGE routines for file processing applications. Consider an intermediate level library to include at least the minimum library, plus a report generator and macros such as table look-up, "put," "get," etc. Consider a complete library to include at least the intermediate library plus functional application packages (e.g., PERT, IMPACT, PRONTO, TRIM, KWIC).		
SS 401	If no library facilities have been included, and/or if provision has not been made for inclusion at a future time, score 0 and omit Specifications SS402-SS406.	yes/no	
SS 402	Is the complete library included? Score 5 for "yes" and omit Specifications SS403-SS406.	yes/no	
SS 403	Is the intermediate level library included? Score 2 for "yes," and answer SS404.	yes/no	
SS 404	Can an intermediate level library be extended to the complete level by the user? Score 1 for "yes" and omit SS405-SS406.	yes/no	
SS 405	Is the minimum library included? Score 1 for "yes" and answer SS406.	yes/no	
SS 406	Can the minimum level library be extended to a higher level? Score 1 for "yes," 0 for "no."	yes/no	

SOFTWARE SPECIFICATIONS

PART 4 - PROGRAM LIBRARY (CONT'D)

SPECI- FICATION NO.	SPECIFICATION	RESPONSE	SCORE
SS 407	Must library programs be written in a language other than the source language? Score 0 for "yes," 1 for "no."	yes/no	
SS 408	Can source language programs be stored in the library? Score 1 for "yes," 0 for "no."	yes/no	

SOFTWARE SPECIFICATIONS

PART 5 - MAINTENANCE, MODIFICATION, AND DOCUMENTATION

SPECI-FICATION NO.	SPECIFICATION	RESPONSE	SCORE
SS 504	Does the documentation include descriptive instruction on how to use the software system and text on the source language? If "yes," score 3 and omit Specifications SS505-SS506.	yes/no	
SS 505	Does the documentation include only descriptive instruction on how to use the software system? If "yes," score 2 and omit Specification SS506.	yes/no	
SS 506	Does the documentation of the software system include only a descriptive text of the source language? Score 1 for "yes."	yes/no	
SS 507	Does the system provide both hard copy documentation and retention of the source program? Score 3 for "yes," and omit Specifications SS508-SS509.	yes/no	
SS 508	Does the system provide only retention of the source program? Score 2 for "yes," and omit Specification SS509.	yes/no	
SS 509	Does the system provide only hard copy documentation of the source program? Score 1 for "yes," 0 for "no."	yes/no	

SOFTWARE SPECIFICATIONS

PART 6 - TRAINING AND FAMILIARIZATION

SPECI-FICATION NO.	SPECIFICATION	RESPONSE	SCORE
SS 601	Are training courses offered by the supplier which cover both the language and practical use of the software system ? If response is "yes," score 3 and omit Specifications SS 602 and SS 603.	yes/no	
SS 602	Are training courses offered by the supplier which cover only the practical use of the software system ? If response is "yes," score 2 and omit Specification SS 602.	yes/no	
SS 603	Are training courses offered by the supplier which cover only the language of the software system ? Score 1 for "yes," 0 for "no."	yes/no	
SS 604	Can the system be used and understood well by appropriate personnel after 2 weeks of "hands-on" familiarization ? Score 3 for "yes," and omit Specifications SS 605 and SS 606.	yes/no	
SS 605	Can the system be used and understood well by appropriate personnel after 3 months of "hands-on" familiarization ? Score 2 for "yes," and omit Specification SS 606.	yes/no	
SS 606	Can the system be used and understood well by appropriate personnel after 6 months of "hands-on" familiarization ? Score 1 for "yes," 0 for "no."	yes/no	

APPENDIX V

INSTALLATION OPERATING SPECIFICATIONS

DIRECTIONS FOR COMPLETING INSTALLATION OPERATING SPECIFICATIONS

The installation operating specifications are designed to gather data to provide statistics and dollar costs for monthly operations that will:

(1) Enable operating management to gauge how it allocates the resources at hand.

(2) Provide input of variable data elements into the multiple regression analyses designed to produce standard indices of numbers and costs.

The collection of resource allocation data is divided into five categories:

(1) Personnel — Statistics on numbers and salary costs for equipment and data preparation operators, control clerks, junior, journeyman and lead programmers, systems programmers, systems analysts, and administrative personnel.

(2) Operations — Statistics on numbers and costs directly related to supplies and forms utilized within the installation.

(3) Equipment Utilization — Statistics related to the utilization of equipment are to be recorded based on hours utilized within specific recording categories. Additional queries relate to the cost of the computer and supporting equipment.

(4) Installation Performance queries are designed to collect data on a project-by-project basis. The data requested relates to the estimated and actually expended man-hours and dollars for each project as well as the delivery time and computer time. A separate sheet should be completed for each project selected on a quasi-random basis.

(5) Programmer Performance queries are designed to provide raw data on how programmers spend their time. The division of the total time into functionally separate categories is intended to provide means for collecting data by project and then grouping these projects by principal activity. Therefore, for each separate Part 4 that is completed, a complementary Part 5 must be completed. Additional quantitative data is required in IS509-512 that will permit correlations of what are assumed to be related facts. It is intended to determine what relationship, if any, exists between the number of problem definition changes and time spent in debugging or documentation, for example.

Wherever recorded data is not available for completion of the specifications, answers based on averages of the values should be indicated with an "A" prefix. Where these values are really judgments based on "questionnaires," they should be noted with a "G" prefix instead of the "A" prefix used to denote the average values. Provision of a procedure for the collection of data will be made available to all installations so that they may adopt uniform data recording procedures.

INSTALLATION OPERATING SPECIFICATIONS

PART 1 - PERSONNEL

SPECIFI-CATION NO.	SPECIFICATION	RESPONSE
IS 101	Number of operators utilized per shift (include tape handlers, peripheral and EAM console operators) Prime 8 hours (need not be between 8.am. - 5.pm.) 1) Following 8 hours of prime shift 2) Next 8 hour period 3) Weekend time 4)	______ ______ ______ ______
IS 102	Total monthly salaries paid to all operators (based on average of previous 6 months salary history)	$______
IS 103	Number of control clerks utilized per shift* (include librarians, batch control, error correction clerks, but not secretaries or receptionists) 1) 2) 3) 4) *Shift as defined in IS 101	______ ______ ______ ______
IS 104	Total monthly salaries paid to all control clerks (based on average of previous 6 months salary history)	$______
IS 105	Total monthly salaries paid to operations and administrative supervisors	$______
IS 106	Number of data preparation operators (keypunch, flexowriter or other key-driven media); if more than 1 shift, provide cumulative total	
IS 107	Total monthly salaries paid to all data preparation operators	$______
IS 108	Total monthly salaries paid to data preparation supervisors	$______
IS 109	Number of junior programmers (trainees with less than 1 year of total experience)	
IS 110	Total monthly salaries paid to junior programmers	$______
IS 111	Number of programmers (journeyman programmers with over 1 year of total experience)	

INSTALLATION OPERATING SPECIFICATIONS

PART 1 - PERSONNEL (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
IS 112	Total monthly salaries paid to programmers	$_____
IS 113	Total monthly salaries paid to lead programmers/ supervisory analysts	$_____
IS 114	Total monthly salaries paid to programming management personnel	$_____
IS 115	Number of maintenance and/or systems programmers (individuals not directly involved with preparation of users programs)	
IS 116	Total monthly salaries paid to systems programmers	$_____
IS 117	Number of systems analysts whose time is chargeable to the computer installation	
IS 118	Total monthly salaries paid to systems analysts	$_____
IS 119	Number of administrative personnel	
IS 120	Total monthly salaries paid to administrative personnel	$_____

INSTALLATION OPERATING SPECIFICATIONS

PART 2 - OPERATIONS

SPECIFICATION NO.	SPECIFICATION	RESPONSE
IS 201	Number of hours of maintenance performed per month	hours
IS 202	Average monthly maintenance costs including spare parts and labor	$_____
IS 203	Mean time between equipment malfunction	hours
IS 204	Mean time to repair equipment malfunction	minutes
IS 205	Number of pages of printed forms utilized per month	pages
IS 206	Average monthly costs for pre-printed line printer forms	$_____
IS 207	Average monthly costs for stock line printer forms	$_____
IS 208	Number of cards used per month	cards
IS 209	Average monthly costs for punch cards	$_____
IS 210	Average number of paper tape reels used per month	reels
IS 211	Average monthly costs for paper tape	$_____
IS 212	Average number of line printer ribbons used per month	ribbons
IS 213	Average monthly costs for console and line printer ribbons	$_____
IS 214	Number of reels of magnetic tape present in the installation	reels
IS 215	Average monthly costs for reels of magnetic tape	$_____
IS 216	Total number of square feet allocated to ADP installation (number includes spare for computer system(s) peripheral equipment, data preparation, tape or card storage, maintenance, receiving and distribution, programmer and operator space)	sq. ft.
IS 217	Charge for cost of ADP installation space expressed as cost per square feet per year	$____/sq.ft.

INSTALLATION OPERATING SPECIFICATIONS

PART 3 - EQUIPMENT UTILIZATION

SPECIFICATION NO.	SPECIFICATION	RESPONSE
IS 301	Total monthly power-on time for each main frame and satellite 1) 2) 3)	_____ hours _____ hours _____ hours
IS 302	Total monthly metered time for each main frame and satellite	_____ hours
IS 303	Total monthly re-run time for total installation Total monthly program test time Total monthly compilation and/or assembly time Total monthly production time	_____ hours _____ hours _____ hours _____ hours
IS 304	Total monthly metered time for a) Magnetic tape units b) Mass storage units c) Card reader d) Paper tape reader e) MICR/Optical/or other document reader f) Card punch g) Paper tape punch h) High speed printer Indicate number of units and total hours.	units hours _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____
IS 305	Monthly computer rental for a) Magnetic tape units b) Mass storage units c) Card reader d) Paper tape reader e) MICR/Optical/or other document reader f) Card punch g) Paper tape punch h) High speed printer i) Supporting EAM equipment	 $_____ $_____ $_____ $_____ $_____ $_____ $_____ $_____ $_____
IS 306	Number of production runs per month	
IS 307	Number of program test runs per month	
IS 308	Number of compilations and/or assemblies per month	

INSTALLATION OPERATING SPECIFICATIONS

PART 4 - INSTALLATION PERFORMANCE

SPECIFICATION NO.	SPECIFICATION	RESPONSE
	For last 50 projects, defined as any job to which a specific charge or identification number is assigned, choose every fifth one (if less than 50 projects, use every third task), indicate the principal activity (see form), and complete the following specifications.	
IS 401	Name of principal activity (Dynamic file processing, static file processing, numeric computation, non-numeric processing, on-line (process) control)	
IS 402	For each principal activity chosen indicate the percentage of the total installation workload which it represents.	
IS 403	Estimated development time (man-hours from the time of problem definition to the time that production capability is reached).	
IS 404	Actual development time (total recorded man-hours from the time of problem definition to the time that production capability is reached).	
IS 405	Estimated completion time for a production system, (hours from the time a job is entered into the installation to time that output is delivered to the user).	
IS 406	Actual completion time for a production system. (Elapsed hours from the time a job is entered into the installation to time that output is delivered to the user).	
IS 407	Estimated computer running time. (Time from start to end of run on a per run per project basis.)	
IS 408	Actual computer running time. (Time from start to end of run on a per run per project basis.)	
IS 409	Was project completed on schedule? If response is "yes," omit Specifications IS 410 and IS 412.	

INSTALLATION OPERATING SPECIFICATIONS

PART 4 - INSTALLATION PERFORMANCE (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
IS 410	Number of days project was completed ahead of schedule. If response is not 0 (zero), omit Specification IS 412.	
IS 411	Number of hours expended (above +, below -) budget	____ ± hours
IS 412	Number of days project was completed behind schedule.	
IS 413	Number of dollars expended (above +, below -) budget	____ ± hours

INSTALLATION OPERATING SPECIFICATIONS

PART 5 - PROGRAMMER PERFORMANCE

SPECIFICATION NO.	SPECIFICATION	RESPONSE
	For those projects cited in the previous section complete the following questionnaire. Indicate the number of hours spent on each area of activity by the 3 levels of programming personnel (junior programmers, programmers - journeyman, lead programmers)	
IS 501	Problem definition/system design	____hours ____hours ____hours
IS 502	Problem analysis/logical analysis (use of a pseudo-language flow chart to product a problem-oriented solution)	____hours ____hours ____hours
IS 503	Machine dependent flow charting (using the source language)	____hours ____hours ____hours
IS 504	Actual source language coding	____hours ____hours ____hours
IS 505	Test data preparation and compilation or assembly of object code	____hours ____hours ____hours
IS 506	Number of checkout runs against prepared test data	____hours ____hours ____hours
IS 507	System test against actual data for integration into the system chain	____hours ____hours ____hours

INSTALLATION OPERATING SPECIFICATIONS

PART 5 - PROGRAMMER PERFORMANCE (CONT'D)

SPECIFICATION NO.	SPECIFICATION	RESPONSE
IS 508	Documentation	____hours ____hours ____hours
IS 509	Total computer time used in Specifications IS 505, 506, 507	____hours ____hours ____hours
IS 510	Supervisors complexity index assigned to this task. Complexity is rated on a scale from 1 to 10 based on 1) non-similarity to previous tasks, 2) degree to which previous coding segments can be utilized, 3) magnitude of total task, 4) number of system logic interfaces.	
IS 511	Number of source language program steps	
IS 512	Number of system design or problem definition changes that occur during the life of the project	

BIBLIOGRAPHY

AUERBACH Corporation. Interim Report on Computer Performance Effectiveness Evaluation Study. 1243-TR-1. Phila. (Feb. 1, 1965).

AUERBACH Corporation. Project 1243 Working Papers. Phila. 1965.

AUERBACH Corporation. Proposal for the Study of Applications, Effectiveness and Problems of the Air Force Information Processing Systems. P-6842. Phila. (Aug. 24, 1964).

AUERBACH Corporation. Quantitative Methods for Information Processing Systems Evaluation. 1134. Phila. (Jan. 1964).

AUERBACH Corporation. Standard EDP Reports. Section 4: Users' Guide; Section 23: Special Reports. Phila., 1965.

Farr, L. & Nanus, B. Factors that Affect the Cost of Computer Programming. ESD-TDR-64-448 (AD 603-707). Mass.:USAF, (July 1964).

Gatto, O. T. "Autosate," Communications of the ACM, 7(6):425-436 (July, 1964).

Gatto, O. T. McGuire, C. B. & VanHorn, R. L. Research Aspects of Command and Control. RM-3155-PR. Calif.: RAND Corp. (Jan. 1963).

Gorn, S. Some Basic Terminology Connected with Mechanical Languages and Their Processors. Phila., University of Pennsylvania (Aug. 1961).

Haverty, J. P. & Patrick, R. L. Programming Languages and Standardization in Command and Control. RM-3447-PR (AD 296-046). Calif., RAND Corp. (Jan. 1963).

Herman, D. J. "The Use of a Computer to Evaluate Computers SCERT," Proceedings of the Fall Joint Computer Conference (AFIPS No. 25):383-395 (Washington, D.C. April, 1964).

Horrigan, T. J. Development of Techniques for Prediction of System Effectiveness. RADC-TDC-63-407 (AD 432-844). New York: RADC (Feb. 28, 1964).

Joslin, E. O. "Cost-Value Technique for Evaluation of Computer Systems Proposals," Proceedings of the Fall Joint Computer Conference (AFIPS No. 25): 367-382 (Washington, D. C.: April, 1964).

LaBolle, V. Estimation of Computer Programming Costs. SP 1747. Calif.: SDC (Sept. 14, 1964).

Minker, J. Trends in Programming: What Should Advanced Development Hardware Expect from Advanced Development Software. Presented at an RCA Meeting, Pennsauken, N.J. (July 23, 1963).

Opler, A. "Measurement of Software Characteristics," Datamation, 10(7):27-30 (July, 1964).

Patrick, R. L. "Measuring Performance," Datamation, 10(7):24-26 (July 1964).

Rosenthal, S. "Analytical Technique for Automatic Data Processing Equipment Acquisition," Proceedings of the Fall Joint Computer Conference (AFIPS No. 25): 359-366 (Washington, D. C.: April, 1964).

US Bureau of the Budget. Guidelines for Appraising Agency Practices in the Management of Automatic Data Processing (ADP) Equipment in Federal Agencies. Circular No. A-61. Washington, D. C.: GPO (Aug. 3, 1963).

US Bureau of the Budget. Report to the President on the Management of Automatic Data Processing in the Federal Government. Washington, D. C.: Senate, 89th Cong., 1st Sess.; Doc. No. 15 (March 4, 1965).

USAF, Electronic Systems Division. RFP for the Study of Applications Effectiveness and Problems of Air Force Information Processing Systems. P. R. No. ES 4-SL-40827, Mass. (July 1, 1964).

USAF. Management Supporting Data Systems. Air Force Regulation No. 300-3. Washington, D. C. (Sept. 1, 1964).

USAF, United States Air Force Data Systems Automation Program, Section III. DSAP-67-3. Washington, D.C. (Jan. 1965).

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

AUERBACH Corporation
1634 Arch Street
Philadelphia, Pennsylvania 19103

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

–

3. REPORT TITLE

AN APPROACH TO COMPUTER INSTALLATION PERFORMANCE EFFECTIVENESS EVALUATION

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

— –

5. AUTHOR(S) *(Last name, first name, initial)*

Statland, N.; Proctor, R.; Zelnick, J.; Getis, R.; Anderson, J.; Taylor, A.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1965	155	5

8a. CONTRACT OR GRANT NO.

AF19(628)-5035

b. PROJECT NO.

Development Planning Study 7990

c.

Task 38

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

1243-TR-2

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

ESD-TR-65-276

10. AVAILABILITY/LIMITATION NOTICES

Qualified requesters may obtain copies of this report from DDC.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Directorate of Analysis, Deputy for Advanced Planning, ESD, AFSC, L. G. Hanscom Field, Bedford, Mass. 01731

13. ABSTRACT

ABSTRACT

Decisions regarding the implementation and evaluation of data processing systems have reflected costly compromises selected from many possible alternatives and there is a need for guidelines which can lead to better decisions regarding automation utilization and planning. The AUERBACH process for computer installation performance effectiveness evaluation operates on a set of specifications and characteristics regarding the principal problem tasks of an installation, its computer complex, and administrative and financial performance.

The process provides objective measures of performance efficiency based on both quantitative and qualitative data, and provides standards for measuring installation effectiveness. Specifications and characteristics are collected via questionnaires, once and only once, in four categories: computer hardware, extended machine (hardware/software interaction), software evaluation and problem specification.

Computer problems can be classified by the environments in which they function, the sources from which data is received and its implied sequence, and the response time within which the computer system is to react. Significant hardware and problem characteristics can be identified, as demonstrated in the AUERBACH VECTOR process (see ESD-TDR-64-194) and estimated running time computed. An extension of this measurement of computer system performance provides a rating for the performance of a given software package on a given piece of hardware by comparing the time derived from the hand-tailored coding to the timing resulting from the object program produced by the software. This ratio measures the efficiency of the software on the specific hardware configuration. The aggregate ratios for all the individual performance criteria are used to derive a performance standard for a software system.

Algorithms are used to summarize the raw data elements and a computer program will select data elements, make simple arithmetic combinations of these elements into composites, and prepare the data for entry into a statistical analysis. Stepwise multiple regression analysis is utilized to determine the relative significance of various data elements and to calculate their relative weights.

DD FORM 1 JAN 64 1473

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computers Computer Performance Computer Effectiveness Software Data Processing						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
________________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
________________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
________________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.